

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Департамент общего образования Томской области
ОГБУ «Региональный центр развития образования»
АНО ДО «Детский технопарк «Кванториум»
Департамент образования администрации г. Томска
МБОУ лицей при ТПУ г. Томска

СБОРНИК ТРУДОВ

XXII Всероссийской конференции-конкурса
исследовательских работ старшеклассников
«Юные исследователи – науке и технике»

26 – 27 марта 2021 г.

Издательство
Томского политехнического университета

Томск 2021

УДК 371.388.6(063)

ББК 74.202.7л0

Ю751

Юные исследователи – науке и технике: сборник трудов XXII Всероссийской конференции-конкурса Исследовательских работ старшеклассников «Юные исследователи – науке и технике»; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2021

В сборнике трудов представлены материалы работ школьников.

Сборник представляет интерес для школьников, занимающихся исследовательской и проектной деятельностью.

В сборник включены статьи, представленные в Оргкомитет конференции и заслушанные на конференции.

СЕКЦИЯ «ИНФОРМАТИКА И РОБОТОТЕХНИКА»

КОМПЬЮТЕРНАЯ ИГРА «ЧАПАЕВ». ЧАСТЬ 2.

Андреев Артемий

МБУДО г. Иркутска ЦДТТ, МАОУ г. Иркутска Лицей ИГУ, г. Иркутск

Руководители: Попов Артём Андреевич, специалист ООО Траснефть-технологии; Рейнгольд Григорий Борисович педагог дополнительного образования.

Данная работа является второй частью работы «Компьютерная игра «Чапаев». В первой части была создана компьютерная реализация советской настольной шашечной игры «Чапаев», было смоделировано нецентральное столкновение шашек, а также создана первая версия автоматического игрока. И еще тогда в планах было сделать версию игры с альтернативными правилами, чему и посвящена вторая часть работы, наряду с созданием второй, более сложной версии автоматического игрока.

Гипотеза: несмотря на изменение правил игры, преимущество в игре останется за игроком, ходящим первым.

Цель: создать версию игры «Чапаев» с альтернативными правилами игры и провести с ней эксперименты, а также создать вторую версию автоматического игрока с более продвинутой стратегией.

Задачи:

1. Разработка новых правил для игры.
2. Разработка второй стратегии для автоматического игрока.
3. Программная реализация вышеперечисленных изменений.
4. Тестирование программы.
5. Проведение экспериментов.

Новизна данной работы заключается в абсолютно новых правилах для данной игры. Во время поисков, нами не было найдено аналогов данной версии игры с альтернативными правилами. Также новизна проявляется в новом автоматическом игроке с отличной от изначальной стратегией.

Был использован язык программирования **Java**, а также среда разработки **IntelliJ** от JetBrains. Чуть ли не в начале работы над изменением правил было решено: изменения коснутся правил передачи хода. Теперь, ход передавался по очереди, в отличии от первоначальной версии игры. Этот этап работы был довольно быстрым, поэтому чуть-ли не сразу началась разработка второй стратегии для автоматического игрока. Созданная стратегия заключалась в переборе всех возможных ходов, а затем

исполнении наилучшего из них. Наилучшим ходом называется тот ход, в котором автоматический игрок сбивает наибольшее количество шашек обычного игрока, но при этом не теряет ни единой своей.

Вскоре началась реализация данных нововведений в самой программе. Изменение правил было довольно легко реализовать, как и саму стратегию автоматического игрока. Однако тут возникла другая проблема: автоматический игрок работал очень медленно. К примеру: в стандартной конфигурации доски 8x8 перебор шел больше восьми минут. К счастью, эта проблема была решена и началось проведение экспериментов с альтернативными правилами игры, по результатам которых во всех конфигурациях доски в режиме с альтернативными правилами игры побеждает игрок, ходящий первым, то есть игрок-человек.

Заключение: в ходе работы был создан режим с альтернативными правилами игры, а также вторая стратегия для автоматического игрока. Также была проведена серия экспериментов с альтернативными правилами игры. Гипотеза подтвердилась.

СПИСОК ЛИТЕРАТУРЫ

1. Чапаев (игра),
[https://ru.wikipedia.org/wiki/%D0%A7%D0%B0%D0%BF%D0%B0%D0%B5%D0%B2_\(%D0%B8%D0%B3%D1%80%D0%B0\)](https://ru.wikipedia.org/wiki/%D0%A7%D0%B0%D0%BF%D0%B0%D0%B5%D0%B2_(%D0%B8%D0%B3%D1%80%D0%B0));
2. Пример реализации импульсного физического движка для моделирования столкновений твёрдых тел.
<https://github.com/aptem336/PhysicsEngine2D>
3. Игра «Чапаев» онлайн, <http://vseigr.net/shashki/2087-igra-chapaev.html>.

СОЗДАНИЕ РУССКОЯЗЫЧНОЙ ВЕРСИИ ИГРЫ PAPER MARIO: THE ORIGAMI KING

Баздырев Александр

МБОУ лицей при ТПУ г. Томска, 10 класс

Руководитель: Буркатовская Юлия Борисовна, канд. физ.-мат. наук,
учитель информатики

В детском возрасте многие видеоигры, в которые я играл, были на английском языке, что сильно затрудняло понимание происходящего на

экране. Тогда я с этим ничего не мог поделать. Но проходили годы, и у меня постепенно нарастал интерес к модифицированию игр. Для меня было очень занятно изменять какие-то элементы игры и потом смотреть на них в действии. Кроме этого я с первого класса дополнительно занимался английским языком, в чём сильно преуспевал. И вот, около трёх лет назад, у меня в голове возникла идея совместить эти два интереса — попробовать себя в переводах игр на русский язык, чем занимаюсь и по сей день. Перевод именно Paper Mario: The Origami King был взят мной сразу после выхода игры (июль 2020) как очередной переводческий проект из-за интереса к игре и ажиотажа вокруг отсутствия перевода к ней в сообществе в социальной сети ВКонтакте компании-разработчика Nintendo.

Цель проекта — создать русскоязычную версию игры Paper Mario: The Origami King в виде скачиваемой модификации (набор модифицированных файлов игры) и выложить её в Интернете в общий доступ. Модификация будет состоять из трёх типов файлов: текстовых (.msbt и .html), изображений (.bfrs) и шрифта (.brfnt). Для редактирования каждого типа файлов используются специально предназначенные для этого программы: Kuriimu и notepad++ — для текста, SwitchToolbox (вместе с Photoshop) — для изображений и FontConverter — для шрифтов. Текст игры перенесён в Google Таблицы и редактируется в них же несколькими переводчиками, включая меня.

Актуальность перевода этой игры довольно высока, поскольку, как я писал выше, в комментариях под записями в официальном сообществе в социальной сети ВКонтакте компании-разработчика Nintendo (и не только там) происходит возмущение по поводу отсутствия локализации игры (перевода на русский язык). Такое недовольство объясняется тем, что средний уровень знания английского языка у населения России не самый лучший. (Россия занимает лишь 36 место из 63 оцениваемых стран по знанию английского языка.) (Лишь 35% опрошенных россиян считают свой уровень английского языка средним (Intermediate).) Следовательно, большинство россиян могут понимать в играх лишь простые фразы и названия, а уловить сложные языковые конструкции, отсылки на известные произведения или игру слов будет для них затруднительно. Это подтверждает высокие актуальность и спрос на подобный проект.

Работа над проектом состоит из восьми этапов: 1) предложение группе переводчиков «FaceOFF» совместную работу над проектом (получение дополнительных переводчиков и помощи в некоторых технических моментах); 2) получение и разбор файлов игры (выяснение, возможно ли

осуществить перевод, подбор программ под каждый тип файлов); 3) создание сообщества в социальной сети ВКонтакте и анонсирование в нём проекта перевода (предварительно создав тестовые файлы шрифта и текста, необходимые для создания снимка экрана игры с демонстрацией перевода), а также рекламирование его в других сообществах по схожей тематике; 4) русификация изображений и трёх шрифтов игры; 5) перенос всего текста игры в Google Таблицы; 5) создание чернового перевода; 6) редакция переведённого текста; 7) тестирование перевода непосредственно в игре; 8) сбор архива с русифицированными файлами игры и выкладывание его в своём сообществе в социальной сети ВКонтакте и группы «FaceOFF» (а также возможное рекламирование его в других сообществах).

Все этапы, кроме пятого и шестого я делаю самостоятельно, лишь иногда прибегая за помощью к техническим специалистам из «FaceOFF», а перевод текста и его редактуру я делаю совместно с несколькими другими переводчиками и редакторами из «FaceOFF». К тестированию, возможно, тоже присоединится ещё несколько добровольцев.

На осуществление вышеперечисленных этапов не потребовались и не потребуются никакие денежные средства, поскольку всё выполняется добровольно. Деньги (4499 рублей) были потрачены разве что на приобретение игры, чтобы извлечь из неё файлы.

На данный момент выполнены уже 1–5 вышеприведённые этапы, то есть завершён черновой перевод. Выполнен он во многих местах некачественно (не благодаря мне), так что редакция предстоит долгая и тщательная.

Этот проект очень трудоёмкий и времязатратный, но найдя себе дополнительных переводчиков из группы переводчиков «FaceOFF», а также создав сообщество в социальной сети ВКонтакте по этому проекту с более 250 людьми (подписчиками), непременно ждущими его завершения, я уверен, что смогу довести его до конца.

ЧАТ-БОТ ДЛЯ СОЦИАЛЬНОЙ СРЕДЫ «ВКОНТАКТЕ»

Бакин Максим

МБОУ лицей при ТПУ г. Томска, 10 класс

Руководитель: Воротов В.Е., учитель информатики

Чат-бот (англ. chatbot) — это программа, которая имитирует реальный разговор с пользователем. Чат-боты позволяют общаться с помощью

текстовых или аудио сообщений на сайтах, в мессенджерах, мобильных приложениях или по телефону.

Актуальность: мы живем в 21 веке, когда множество действий происходит в онлайн, независимо от человека. И чат-боты это одна из основ социальных сетей. Они позволяют взаимодействовать с людьми сразу же после того, как они написали. Человеку не нужно находиться на месте администратора. Достаточно создать бота, который будет отвечать заготовленными фразами на определённые вопросы. И отвечать он может делать одновременно нескольким людям, что существенно облегчает работу и уменьшает время обслуживания.

Цель: Изучить принцип создания чат-ботов, их сильные и слабые стороны и создать подобного.

Задача: Изучить принцип работы и создания чат-ботов, научиться работать с сервером, найти место для сервера и установить туда бота, создать группу “Вконтакте” для публичного доступа к нему.

Этапы исследования:

1) Исследование подходящего языка программирования/программ для создания.

2) Исследование рынка, определиться с набором команд, на которые будет отвечать бот, выбрать для него тематику.

3) Создание чат-бота на основе полученных знаний.

4) Тестирование его в общедоступной группе.

После тестирования бота станет понятным, какие проблемы следует устранить и можно будет оценить его работоспособность, для улучшения и развития проекта.

Заключение. В ходе работы я выяснил, что подобные боты всегда актуальны, т.к. количество групп в социальных сетях постоянно растет (например: группа магазина одежды). И многим из них нужны такие боты. Сам опыт исследования этой темы развивает мои навыки программирования, работы с соц. сетями и аудиторией. Созданный чат-бот смог отвечать на вопросы, установленные как шаблон и взаимодействовать с пользователем встроенными в нем командами. Я получил опыт в создании бота и продолжу добавлять в него функционал и улучшать скорость его работы, улучшая код, на котором работает данный бот.

СПИСОК ЛИТЕРАТУРЫ

1. <https://vk.com/dev/CommunityMessages>
2. <https://sendpulse.com/ru/support/glossary/chatbot>

3. <https://habr.com/ru/post/428507/> (помощь с кодом).
4. <https://codecamp.ru/blog/python-python-networking/>.

ТРЕНАЖЕР ПО БИОЛОГИИ «СДАМ ЕГЭ»

Вельгер Вячеслав,

Муниципальное общеобразовательное учреждение «Средняя школа № 18», 11 класс, г. Ачинск

Руководители: Жерносек О. Н. учитель информатики, Сидоренко С.В.
учитель биологии

Актуальность. Начиная с 2017 по 2020 г.г. экзамен по биологии претерпевал изменения в структуре заданий, первая часть - вопросы с № 1 по № 21, где нужно дать краткий ответ или выбрать несколько вариантов из предложенных, были существенно изменены и дополнены заданиями нового формата. В первой части заданий ЕГЭ не осталось вопросов с одновариантным выбором ответа, но появились такие, где надо выбрать, например, два варианта из пяти. И это тоже указано в формулировке вопроса - важно это прочитать и обратить внимание [1]. Наличие знаний по предмету биология еще не гарантирует успешную сдачу экзамена. Навыки правильного прочтения и решения заданий обучающиеся отрабатывают с учителем, учитель контролирует внимательность прочтения формулировки задания и, производит корректировку понимания того, что требуется от обучающегося.

При самостоятельной подготовке к экзамену обучающийся лишен этой помощи или лишен живого общения с учителем, используя электронные платформы. В заданиях, где есть выбор ответов, прежде чем выбрать из предложенных, обучающийся отрабатывает данный навык с учителем на первых этапах подготовки, самостоятельные ответы обучающегося на вопрос проверяются учителем, ошибки прорабатываются, совместно подбираются наиболее подходящие или совпадающие варианты. При таком приеме вероятность ошибки резко падает. При самостоятельной подготовке к экзамену обучающийся должен самостоятельно проработать ошибки, подобрать наиболее подходящие или совпадающие варианты, порою это крайне затруднительно, так как предложенные ответы на задания являются неполными, не указывают на элементы допущения неточной интерпретации биологической информации.

Постановка и формулировка проблемы. Для самостоятельной подготовки к ЕГЭ по биологии в сети Интернет существует множество пособий, но они, в основном, используются учителями, а не учащимися [2]. Как правило учитель при подготовке к ЕГЭ распечатывает задания на бумажных носителях, что крайне неудобно и затратно, а тренажеры, при помощи которых можно самостоятельно решать задания, не всегда доступны обучающимся, так как располагаются на разных платформах и контенте, имеющие ответы к заданию без алгоритма решения. Сайты, направленные на подготовку к ЕГЭ, не всегда содержат обновленные задания по измененной структуре экзамена [3]. Традиционно самые сложные разделы в биологии - те, что являются наименее наглядными.

В связи с распространением Covid 19 в 2020-2021 г.г. обучающимся все чаще приходится пользоваться электронными пособиями для подготовки к экзамену, использовать информацию различных сайтов, контентом, электронных платформ. Это не всегда удобно для самостоятельной подготовки, потому что крайне сложно ориентироваться в спецификации экзаменационного материала.

Именно эти проблемы легли в основу создания электронного пособия для самостоятельной подготовки к экзамену с использованием алгоритмов и четко структурированного материала в соответствии с кодификатором и спецификатором экзаменационного КИМа по биологии.

Разработанность исследуемой проблемы. Существует немало платформ, сайтов, которые содержат тренировочные задания по биологии, но в них не всегда приводится алгоритм решения, инструкции, полезные аспекты решения и план выполнения заданий. На некоторых платформах не приведено изменение заданий № 27-матричный биосинтез, который был введен в структуру ЕГЭ с 2020 года [4;5]. Большинство сайтов, вебинаров, разнообразных контентом и платформ содержат платные условия работы, бесплатным является только регистрация, просмотр главной страницы, иногда существует возможность использования небольшого объема информации на бесплатных условиях [6;7].

Гипотеза: можно создать электронное пособие как доступный практический тренажер для самостоятельной отработки навыков решения заданий при подготовке к ЕГЭ по биологии.

Цель: создание электронного пособия-тренажера, направленного на самостоятельную подготовку к Единому государственному экзамену по биологии (ЕГЭ).

Задачи:

1. Подобрать теоретический и практический материал в соответствии с измененной структурой заданий ЕГЭ по биологии.
2. Сформулировать разделы электронного пособия-тренажера.
3. Выбрать среду разработки.
4. Создать электронное пособие-тренажер по биологии.
5. Презентовать электронное пособие-тренажер обучающимся и учителям.
6. Провести опрос и обработать полученные результаты.

Методы исследования: аналитический, информационное моделирование, компьютерное моделирование, метод обработки статистических данных.

Используя аналитический метод, был отобран теоретический, практический материал, алгоритмы работы по всем видам заданий 1-28 из структуры КИМа ЕГЭ по биологии, инструкции, полезные аспекты решения, план выполнения заданий, примеры решения, оценки требуемых от обучающегося умений, подобрали и распределили по уровню сложности задания для проверки знаний по данной теме.

При использовании информационного моделирования, были составлены разделы электронного пособия-тренажера: «Алгоритмы работы», «Практика», «Дополнительный материал».

С помощью компьютерного моделирования, разработал электронное пособие-тренажер, используя среду разработки Unity и язык программирования C#[8].

Этапы создания электронного пособия-тренажера по биологии.



Рисунок 1. «Новая сцена»

Первым этапом создал новую сцену (Рисунок 1).

Вторым этапом создал раздел «Алгоритмы работы» (Рисунок 2), который содержит подробный алгоритм, аспекты решения заданий линии 1-19, 20-22, 23-26 и 27-28 (Рисунок 3)

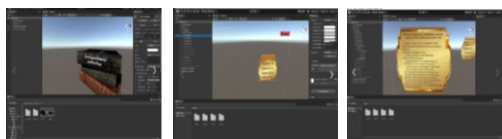


Рисунок 2. «Раздел «Алгоритмы работы»



Рисунок 3. «Алгоритм, аспекты решения заданий линии 1-19, 20-22, 23-26 и 27-28»

Третьим этапом создал раздел «Практика», содержащий задания в соответствии с кодификатором и спецификатором ЕГЭ по биологии

(Рисунок 4).



Рисунок 5. «Дополнительный материал»



Рисунок 4. «Практика»

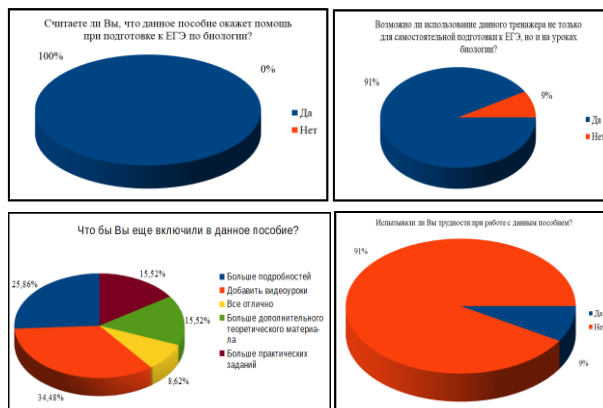
Четвертым этапом создал раздел «Дополнительный материал», содержащий экологическую информацию, закон Хайди Вайенберга, примеры решения задач по популяционной генетике и информацию про жизненные циклы живых организмов (Рисунок 5).

Пятым этапом написал код программы на языке программирования C#.

В течение урока обучающиеся 10-11 классов работали с тренажером, используя разделы «Алгоритмы работы», «Практика» и «Дополнительный материал». По окончании работы с электронным пособием-тренажером обучающимся было предложено ответить на следующие вопросы:

1. Считаете ли Вы, что данное пособие окажет помощь при подготовке к ЕГЭ по биологии?
2. Возможно ли использование данного тренажера не только для самостоятельной подготовки к ЕГЭ, но и на уроках биологии?
3. Что бы Вы еще включили в данное пособие?
4. Испытывали ли Вы трудности при работе с данным пособием?

Результаты опроса были обработаны и представлены в виде диаграмм, которые можно видеть ниже:



В результате анкетирования данный тренажер вызвал интерес у обучающихся сдающих ЕГЭ по биологии.

Заключение. Тренажер удобен в использовании и не требует материальных затрат. Имеет нужную информацию для того, чтобы дать ответы на поставленные вопросы. Тренажер по биологии – «Сдам ЕГЭ» можно использовать для дистанционного обучения, индивидуальных занятий, а также самостоятельной подготовки к ЕГЭ по биологии. Следовательно, была подтверждена выдвинутая гипотеза, достигнуты цель и поставленные задачи.

Работа над пособием будет продолжена. Тренажер будет дополнен видеороликами, которые помогут более качественной самостоятельной подготовке, а также его использованию на уроках биологии, естествознания при закреплении определенных тем и разделов, которые лишены наглядности.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный институт педагогических изменений: <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!/tab/151883967-6>
2. Учительский портал: <https://www.uchportal.ru/load/75>
3. Незнайка: <https://neznaika.info/ege/biology/>
4. Решу ЕГЭ: <https://bio-ege.sdangia.ru/>
5. 4егэ: <https://4ege.ru/video-biologiya/>
6. ЕГЭ box биология. Онлайн-подготовка к ЕГЭ с гарантией поступления: <https://egebox.ru/courses/11/bio/>
7. Фоксфорд: <https://express.foxford.ru/>
Unity (игровой движок) - <https://ru.wikipedia.org/wiki/Unity>

УМНЫЙ ДОМ

Вершинин Никита

МБОУ Лицей № 3 г. Иркутска, 11 класс

Руководитель: Вайншенкер Анна Борисовна, учитель информатики и ИК

На сегодняшний день умный дом довольно популярен, но из-за дороговизны и из-за недостатка информации на русском языке установка сервера умного дома проблематична.

Различные способы связи и IT технологии стремительно развиваются, в данный момент многие бытовые приборы могут общаться с окружающими

его устройствами с помощью различных способов связи (rf 433mhz, wi-fi, мобильная сеть, bluetooth, 2.4Гц), но мало кто пользуется этим удобством. Мне не хотелось каждый день ходить в теплицу и поливать грядки, но умный дом может с помощью датчиков определить, когда нужно включить полив и делать это в автоматическом режиме. Также, если я нахожусь вне дома, он сможет сообщить мне о протечке или пожаре в доме, или о том что кто-то пришел, отправляя при этом фотографию этого человека. Профессиональные службы охраны делают свою работу хорошо, но часто за большие деньги, а тут в зависимости от ситуации можно самому решить, что лучше сделать. В умном доме можно настроить сцены, это мини-программы, которые могут сделать к примеру освещение для просмотра кино или выключить везде свет.

Именно поэтому целью моего проекта стало создание инструкции и проведение мастер-класса для учеников, которые заинтересованы в создании умного дома и уже имеют начальные знания систем Linux.

Цель проекта: Создать инструкцию по установке сервера умного дома и настройке панели управления домом для единого управления за домашними системами для людей, которые хотят сделать собственный умный дом.

Задачи проекта:

1. Изучить теорию по настройке систем Linux
2. Выбрать комплектующие для создания умного дома исходя из характеристик
3. Установить на одноплатный компьютер сервер и второстепенные программы для создания умного дома и панели управления
4. Написать программы для управления модулями умного дома на базе Arduino и чипов ESP
5. Синхронизировать модули с сервером умного дома с помощью Wi-Fi, Bluetooth и других способов связи
6. Собрать материал для оформления.

Для установки сервера нам понадобится одноплатный компьютер и micro-sd флешка на 8-32Гб.

Для начала нужно скачать систему для нашего компьютера. У меня orange pi pc, поэтому я зашел на сайт <https://www.armbian.com/orange-pi-pc/> и скачал последний образ Ubuntu 19.04 LTS Bionic. Также нам нужна утилита balenaEtcher, с помощью нее мы запишем образ системы Ubuntu на

micro-sd (прил.1). После записи нужно вставить флешку в одноплатный компьютер и подключить к нему ethernet кабель, чтобы не искать монитор и клавиатуру для него. Далее нужно зайти в настройки роутера и узнать ip-адрес одноплатного компьютера(прил.2), скачать Pally для подключения через протокол ssh и настроить на подключение к компьютеру. Установить рут-пароль, настроить своего пользователя(прил.3) и обновить систему командами:

- *sudo apt-get update*
- *sudo apt-get upgrade*

На одноплатный компьютер нужно установить программы, без которых мы не сможем установить сервер Home Assistant (docker.io, bash, jq, curl, avahi-daemon, dbus). Команда для их установки:

- *sudo apt-get install docker.io bash jq curl avahi-daemon dbus*

После чего скачать и запустить скрипт для установки сервера Home Assistant этой командой:

- *curl -sL https://raw.githubusercontent.com/home-assistant/hassio-installer/master/hassio_install.sh | bash -s -- -m qemuarm*

Пока устанавливался сервер Home Assistant, можно зайти через браузер по адресу <http://192.168.1.170:8123/> (ip моего одноплатного компьютера с портом 8123), и увидеть приветственную заставку (прил.4)

Далее можно установить сервер Mosquitto (MQTT брокер для связи с модулями) по этой инструкции:

1. *Выполнить команды в терминале:*
 - *sudo apt-get install mosquitto mosquitto-clients*
 - *sudo mosquitto_passwd -c /etc/mosquitto/passwd ИМЯ ПОЛЬЗОВАТЕЛЯ*
 - *sudo nano /etc/mosquitto/mosquitto.conf*
2. *И написать в конце файла:*
 - *allow_anonymous false*
 - *password_file /etc/mosquitto/passwd*
3. *Выполнить команду:*
 - *sudo systemctl restart mosquitto*

После подождать около 20 минут (это время занимает установка сервера умного дома) и в конце мы увидим, что нам предлагают создать пользователя умного дома, об этом уже в следующем пункте

Настройка сервера умного дома

Первый этап - задать имя пользователя, а также его логин и пароль (прил.5)

После него нас спросят, хотим ли мы сразу добавить интеграции, но на данном этапе мы пропустим добавление интеграций.

Сразу попадаем на начальную страницу lovelace-ui (интерфейс сервера), нажимаем в нижнем левом углу на аватарку пользователя (обычно 1 буква имени), меняем язык на нужный нам.

После переходим в настройки, заходим в раздел опции и ставим нужные данные(название, где находится дом, часовой пояс и высота)

Далее заходим в настройки>интеграции, нажимаем справа "+", пишем в поиске mqtt и нажимаем на него, заполняем поля: в брокер пишем ip адрес одноплатного компьютера, порт не меняем, и пишем логин и пароль которые создали при установке mqtt.

На этом первоначальная настройка сервера умного дома закончена. Чтобы подключить модуль или технику к умному дому, нужно зайти в настройки Home Assistant'a, далее зайти в интеграции и выбрать нужную интеграцию. Это может быть телевизор, кондиционер, модуль ESP и многое другое.

Также можно прописать нужные модули в конфиге, это нужно делать с помощью дополнения File Editor в Home Assistant. Там можно найти файл configurations.yaml и добавить датчики с помощью параметра sensor и реле с помощью параметра switch.

В итоге файл configurations.yaml получился таким:

```
default_config:
homeassistant:
  name: Sim$ka
  time_zone: Asia/Irkutsk
  unit_system: metric
  customize: !include_dir_merge_named includes/customize
  auth_providers:
```

```
- type: homeassistant
- type: legacy_api_password
  api_password: !secret http_pass
whitelist_external_dirs :
- /config
sensor:
- platform: mqtt
  name: "TemperatureG"
  state_topic: "hass/sensor/garage/temperature"
  unit_of_measurement: "°C"
switch:
- platform: mqtt
  name: "FAN1"
  command_topic: "hass/switch/lr/fan1"
  state_topic: "hass/switch/lr/fan1/state"
```

Создание правил автоматизации

Автоматизации нужно задавать в настройки > автоматизации. Тут мы можем создать автоматизацию по времени, состоянию датчика, значению датчика и т.д на любой сенсор или реле.

В автоматизациях есть:

Триггеры (прил.6) - это то, что запускает процесс автоматизации. Можно указать несколько триггеров на одно и то же правило. Когда триггер работает, Home Assistant будет проверять условия (если таковые имеются), и выполнять действия.

Условия (прил.7) - это необязательная часть правила автоматизации. Действие автоматизации не будет выполнено, пока не будут удовлетворены все условия.

Действия (прил.8) - это то, что сделает Home Assistant, когда правило автоматизации работает.

Заключение

В процессе моей работы над проектом я узнал, как работать с системой Linux, как установить сервер Home Assistant, настроить его, а также подключить модули ESP к серверу умного дома.

IT технологии развиваются с огромным темпом, и можно с уверенностью сказать, что умные дома в скором времени будут появляться все чаще и чаще, поэтому я считаю свой проект актуальным. У сервера Home Assistant очень много плюсов, например, возможность подключения более 3000 устройств, при этом не имея особых знаний программирования.

У меня получилось создать инструкцию для установки и настройки сервера и модулей ESP, поэтому **мой проект можно считать завершенным**. Но я буду продолжать работу над этим проектом, и планирую максимально автоматизировать дом.

Я выделил самые главные полученные знания в работе со своим проектом:

- Знания систем Linux и написание команд в командной строке
- Установка сервера умного дома и дополнительных серверов
- Настройка сервера умного дома и MQTT
- Написание программ для модулей ESP и подключение к ним датчиков и реле
- Подключение бытовой электроники и модулей ESP к серверу умного дома

Уверен, что эти знания мне очень пригодятся в моей будущей профессии.

Итак, я купил одноплатный компьютер и модули к нему, установил систему Linux, на нее установил сервер Home Assistant, дополнил его сервером MQTT, настроил и подключил модули к серверу и написал инструкцию для людей, имеющих начальный уровень знаний систем Linux и с желанием сделать свой собственный умный дом или квартиру.

СПИСОК ЛИТЕРАТУРЫ

1. Валади, Дж. 100% самоучитель Linux / Дж. Валади. - М.: Технолоджи-3000, 2018. - 336 с.
2. Колисниченко, Д.Н. Linux. Полное руководство / Д.Н. Колисниченко, Аллен, Питер В.. - М.: СПб: Наука и Техника, 2017. - 784 с.
3. Колисниченко, Д.Н. Ubuntu Linux 7.04. Руководство пользователя (+DVD) / Д.Н. Колисниченко. - М.: СПб: Питер, 2016. - 189 с.
4. Кюнель Samba: интеграция Linux/Unix-компьютеров в сети Windows / Кюнель, Йенц. - М.: Мн: Новое знание, 2016. - 399 с.

5. Костромин В.А. Основы работы в ОС Linux. Курс ИНТУИТ.ру. - www.intuit.ru
6. Костромин В.А., Разделы диска и средства для работы с ними в Линукс. - <http://www.linuxcenter.ru/lib/books/partitioning/>
7. Корнеев Д. Права доступа к файлам в Linux - <http://old.linux.kiev.ua/modules.php?name=News&file=sid=703>
8. Кузнецов С.В. - История создания UNIX - http://www.linuxcenter.ru/lib/history/unix_gentree.phtml

ИГРОВЫЕ ПРИЛОЖЕНИЯ С СЕНСОРНЫМИ ПЕРЧАТКАМИ В СРЕДЕ UNITY ДЛЯ РЕАБИЛИТАЦИИ ПОСЛЕ ИНСУЛЬТА

*Москаленко Александр, Гордеев Кирилл,
Обходимов Андрей*

МАОУ КУГ “Универс” № 1, 11 класс

Руководитель: Вахитова Е. Ю., учитель информатики

Цель:

- Создание пула игровых приложений для восстановления после инсульта с сенсорными перчатками в среде Unity.

Задачи:

- Изучить методические рекомендации для восстановления мелкой моторики рук
- Изучить документацию по работе с сенсорными перчатками
- Разработать управление в играх согласно методическим предписаниям и исходя из возможностей сенсорных перчаток
- Изучить теоретические основы работы в среде Unity 3D
- Подключить сенсорные перчатки в качестве манипуляторов
- Создать концепты игр в среде Unity 3D
- Протестировать на группе целевой аудитории удобство и продуктивность приложений
- На основе тестирования реализовать игры
- Апробировать готовые игры

Основная часть

Создание концептов игр в среде Unity

Основные движения и их примерное использование были выбраны, оставалось придумать основную суть игры. Так как движения кисти должны отвечать за передвижения игрока, идеальным выбором было взять за основу игру про гонки.

Учитывая средний возраст пациентов кому придется пользоваться пулом игр, выбор второго приложения пал на старую игру Battle city. В ней главным игроком выступает танк. В итоге было принято решение сделать игру, в которой от пользователя будет зависеть только движение танка от третьего лица. Так как приложением будут пользоваться люди восстанавливающиеся после болезни, необходимо упростить управление до минимума, поэтому наведение пушки и выстрелы будут происходить автоматически.

Третьей игрой стала Beat Saber, суть которой заключается в прохождении уровня путем разбиения летящих на персонажа 3d фигур разного цвета мечом под интересную мелодию.

Когда идеи были подготовлены, оставалось создать сами концепты. Первым готовым приложением стала гоночная игра ArmRacing. Используя возможности среды Unity, были использованы компоненты Rigidbody, Mesh Collider, AudioSource и многие другие. Для реализации движения автомобиля были созданы отдельные скрипты отвечающие за отслеживание столкновений, сбора очков и разгона. За сохранение рекордов и настроек персонажа отвечают также отдельные скрипты, задача которых локально хранить информацию, которая может в будущем понадобится для лаунчера заказчика.

Вторым концептом, над которым была завершена работа стала игра - Battle of Tanks, основой для которой послужила старая игра - Battle city. В создании игры использовались такие же компоненты, как и в ArmRacing, но также необходимо было реализовать автоматическое наведение орудия танка, выбор ближайшей цели, и выстрел в определенный момент. Для реализации перечисленных задач использовалась дополнительная технология Unity, такая как RayCasting. Её суть заключается в отслеживании определенного луча, испускаемого объектом. В данном случае - это луч, выпускаемый из орудия танка. Алгоритм нахождения ближайшего врага необходим для успешного выживания героя. Так как снаряд также имеет свою физику, из-за этого выстрелы на дальней дистанции могут быть не точны. Выстрел происходит в определенный момент, когда орудие танка было успешно перезаряжено, а луч столкнулся с объектом врага. Для осуществления выстрела пользователю не нужно

предпринимать дополнительных действий. Его задача - только в управлении танком с помощью движений перчаткой.

В игре Beat Saber необходимо было реализовать движение меча, при помощи которого пользователь мог бы разбивать кубики. Для простоты управления мечом стала кисть играющего человека. Таким способом получилось облегчить управление и к тому же увеличить его продуктивность. Для реализации всей этой идеи использовалась также технология RayCasting и основные компоненты среды Unity.

Заключение

Результат

Три готовых игры с использованием сенсорных перчаток в качестве манипуляторов для реабилитации после инсульта.

Выводы:

- Изучены методические рекомендации для восстановления мелкой моторики рук
- Изучены теоретические основы работы в среде Unity 3D.
- Разработано управление в играх согласно методическим предписаниям и исходя из возможностей сен-сорных перчаток
- Изучена документация с сайта senso.me для сенсорных перчаток
- Созданы концепты в среде Unity 3D
- Подключены сенсорные перчатки в качестве манипуляторов
- На данный момент ведутся переговоры о добавлении игр в пул медицинских игр от компании SensoMe и дальнейшее их тестировании уже на настоящих пациентах.

ПРИЛОЖЕНИЕ ДЛЯ БЕЗОПАСНОЙ ПЕРЕДАЧИ ДАННЫХ НА ОСНОВЕ АЛГОРИТМА TripleDES

Горст А.Н.

*Муниципальное бюджетное общеобразовательное учреждение «Лицей
при ТПУ», 10 Класс.*

С развитием технологий и массовым внедрением их в повседневную жизнь, проблема безопасности, безопасной передачи данных в интернет пространстве становится одной из самых приоритетных задач. Для безопасной передачи данных используются различные алгоритмы шифрования данных, как симметричные, так и асимметричные. Одним из самых распространенных алгоритмов является TripleDES - быстрый и достаточно криптостойкий алгоритм.

Принцип работы симметричных шифров: для шифрования и дешифрования состоит в том, что применяется один и тот же криптографический ключ, который должен храниться в секрете обеими сторонами.

Преимущества симметричного шифрования по сравнению с асимметричным:

- Скорость (в 3 раза быстрее асимметричных).
- Меньшая требуемая длина ключа для сопоставимой стойкости.
- Изученность.

Недостатки:

- Сложность обмена ключами.
- Сложность управления ключами в большой сети.

Ниже представлена схема шифрования.

Схема шифрования

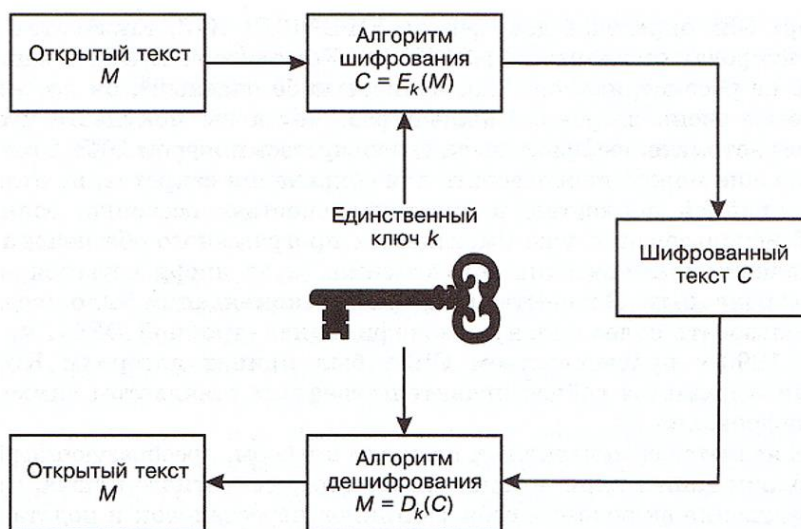


Рисунок 1 – Схема шифрования

Блочные шифры используют один и тот же алгоритм шифрования для каждого блока. Из-за этого блок обычного текста всегда будет возвращать один и тот же зашифрованный текст при шифровании одним и тем же ключом и алгоритмом. Поскольку это поведение может быть использовано для взлома шифра, вводятся режимы шифрования, которые изменяют процесс шифрования на основе обратной связи от более ранних блочных шифров. Данная модификация обеспечивает более высокий уровень безопасности, чем простое блочное шифрование.[3]

Целью настоящей работы является создание удобного приложения с понятным интерфейсом, позволяющее осуществлять шифрование/дешифрование данных и их передачу на основе блочного симметричного шифра TripleDES, с возможностью использования различных режимов шифрования и видов дополнения блоков.

Приложение типа WindowsForms (.NET Framework) разрабатывалось в среде разработки IDE Microsoft Visual Studio 2017, язык программирования – VisualC#, при этом использовались следующие библиотеки: System.Security.Cryptography (Работа с алгоритмами шифрования) [2], System.IO (Работа с потоками), System.Windows.Forms (Работа с пользовательским интерфейсом).

Этапы выполнения:

- Изучить работу современных алгоритмов шифрования, подобрать оптимальный.
- Реализовать приложение.
- Реализовать инструкцию (документацию) по приложению.

Пользователь получает возможность зашифровывать текстовые данные алгоритмом TripleDes, самостоятельно выбирая режимы шифрования, а также передавать автоматически созданный на рабочем столе файл другому пользователю, который, в свою очередь, имеет возможность расшифровать информацию. При этом данная информация непонятна посторонним людям, не имеющим исходного ключа. Например, текст «Hello World» в зашифрованном виде выглядит как «DŸ[.ĥ ъX(‡а%о)-¶».

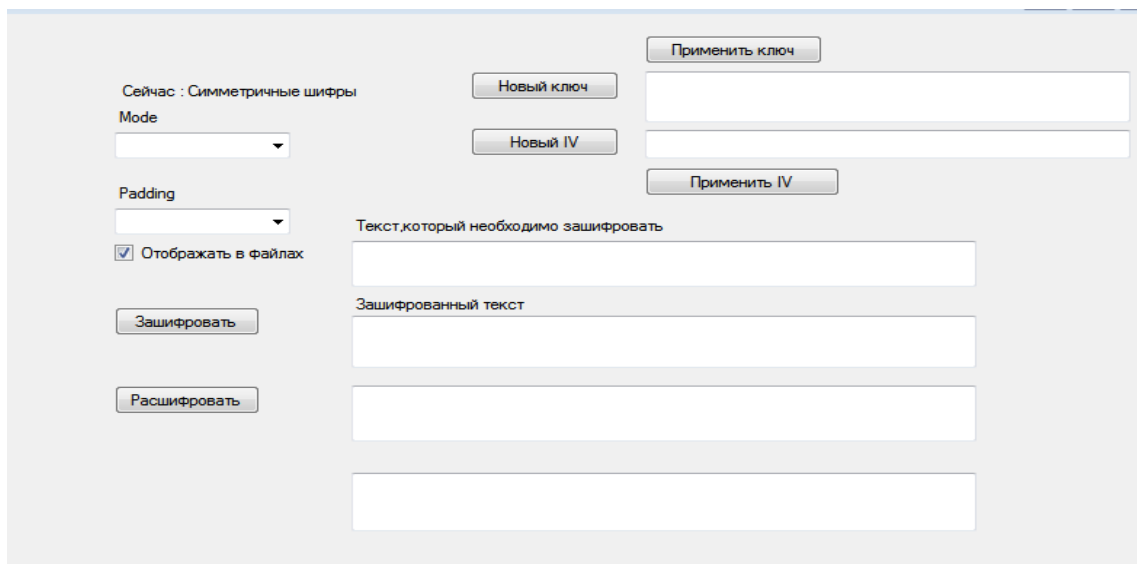


Рисунок 2 – Интерфейс приложения

Нам удалось создать полноценное приложение, работающее на основе блочного симметричного шифра TripleDES. Реализованы различные операционные режимы шифрования: электронная шифровальная книга (ECB), сцепление шифрованных блоков (CBC), шифрованная обратная связь (CFB), обратная связь по выходу (OFB) [3], а также следующие виды дополнения блоков: PKCS7, Zeros, None. [3]

Стоит отметить, что не все режимы доступны на всех операционных системах. Однако, набор реализованных операционных режимов позволяет пользоваться приложением на большинстве операционных систем Windows.

СПИСОК ЛИТЕРАТУРЫ:

1. Руководство по языку C#.URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp>. (дата обращения: 13.03.2021).
2. Руководство по работе с CryptoStream, C#. URL: <https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.cryptostream?view=netframework-4.8> (дата обращения: 13.03.2021).
3. Документация по работе с операционными режимами симметричных шифров, C#.URL:<https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.ciphermode?view=netframework-4.8> (дата обращения:13.03.2021).
4. Сингх Саймон, Книга шифров .Тайная история шифров и их расшифровки.

УМНАЯ ТЕПЛИЦА «КОРМИЛИЦА»

Дранов Денис Анатольевич

*Муниципальное автономное общеобразовательное учреждение
«Средняя общеобразовательная школа №7» г. Колпашево Томской области*

В последнее время наблюдается особый интерес к робототехнике и автоматизированным системам во всех областях народного хозяйства.

Я занимаюсь робототехникой шестой год. Сначала я создавал и программировал роботов в основном для соревнований, а теперь мне нравится изучать и проектировать робототехнические системы для использования в быту.

Колпашевский район – это в основном сельская местность, где почти все жители занимаются выращиванием сельскохозяйственных культур на дачных участках.

Передо мной возникла проблема: невозможность своевременного соблюдения благоприятных условий для роста и созревания томатов в открытом грунте на нашем мичуринском участке.

Я предположил, что посильную помощь в этом может оказать «умная теплица», обустроить которую вполне можно своими руками и без чрезмерных затрат.

Исходя из этого, я поставил перед собой **цель**:

спроектировать и создать макет «умной теплицы» для комфортных условий роста и развития растений, для облегчения труда по выращиванию теплолюбивых овощных культур с применением информационных технологий, для ознакомления с принципом ее работы, с последующим применением данной системы в жизни.

Для достижения этой цели мне необходимо будет **решить ряд задач**:

- подобрать необходимые устройства и оборудование для работы,
- определить сферу применения «умной теплицы» и условия ее работы,
- сконструировать макет,
- запрограммировать контроллер Arduino для автономной работы.

Объектом исследования являются характеристики комплектующих, конструктивные особенности автоматизированных систем и особенности современных «умных теплиц».

Предмет исследования – автоматизированная система для выращивания томатов «умная теплица» «Кормилица».

Теоретическая значимость заключается в том, что найденные мною технические характеристики и составленная таблица могут помочь правильно использовать изделие, выполненное из электронных компонентов, в разных хозяйствах.

Практическая значимость измеряется в прикладной ценности полученных результатов, а также в самой идее – сделать самостоятельно макет «умной теплицы».

Итак, в ходе работы над темой исследования мною было изучено 7 электронных источников и 5 печатных, в которых я нашел материал, характеризующий особенности предмета исследования.

Результаты исследования показали:

Для томатов важны условия их выращивания, правильный температурный режим и влажность. При температуре ниже 10°C растения могут погибнуть, а при температуре выше 35°C не завязываются плоды.

По данным метеорологической станции г. Колпашево температура у нас не создает благоприятных условий для выращивания томатов в открытом грунте, особенно вредны ранние заморозки.

Следовательно, наилучшим «домом» для помидоров является теплица, т.к. даже простейшая пленочная конструкция помогает повысить урожай в 2-4 раза по сравнению с тем, что можно получить, высадив томаты в грунт.

После анализа рынка мы отметили несколько торговых фирм (на слайде – две теплицы). Основываясь на карты покрытия сотовой связью нашего района и учитывая уровень доходов населения (на слайдах – карты покрытия и данные о доходах), мы выявили у данных теплиц ряд недостатков:

- обмен информации осуществляется через wi-fi (для дачи это нереально)
- не во всех дачных поселках есть сигнал GSM
- стоимость комплекта не всем по карману
- работает от электрической сети, значит необходимо протягивать электропровод по дачному участку

Перед конструированием и программированием системы мы провели экономические расчеты стоимости комплекта для нашей умной теплицы. По результатам которых мы получили следующее: все комплектующие имеются в магазинах, стоимость теплицы – низкая и приемлема для нашего населения, не используется сотовая связь и Wi-Fi (слайды с таблицами).

Мы учли недостатки имеющихся теплиц и остановились на выборе металлического каркаса в виде домика с поликарбонатным покрытием с вентилятором, работающим от солнечной батареи. Это позволит использовать электроэнергию на дачном участке и в других целях. Наша теплица может быть оснащена инфракрасным обогревателем, с целью дополнительного нагрева и решения проблемы повышенной влажности воздуха (если будет такая необходимость), разводкой системы полива (трубы), подключенной к ёмкости с водой с датчиком уровня воды, для увлажнения почвы. Мы определились с сельскохозяйственной культурой – томаты, которая будет выращиваться только в летнее время и в условиях, приравненных к крайнему северу. У нас в летнее время долгота светового дня достигает 18 часов, что благоприятно влияет на рост, развитие и плодоношение данной культуры.

Для нашей умной теплицы мы сделали упрощенный макет, который включает: контроллер, датчик температуры и влажности воздуха, датчик влажности почвы, погружной насос и ёмкость с водой. Это действующая система автополива в зависимости от температуры воздуха и влажности почвы. Наша умная теплица «Кормилица» работает по следующему алгоритму:

- влажность воздуха $< 20\%$ и $t > 40^\circ$ или влажность грунта $< 8\%$, то включается насос для полива и выключается вентилятор, пока не станет влажность 50% ;

- влажность $> 50\%$ и $< 80\%$, $t > 40^\circ$; то включается вентилятор;

- влажность $> 80\%$, то включается вентилятор и включается обогреватель;

- $t < 10^\circ$; то включается обогреватель и выключается вентилятор.

Показ – демонстрация работы

Сейчас я симитирую неблагоприятные условия для растений. Феном я поднимаю температуру воздуха > 30 градусов и автоматически включается вентилятор.

На жидкокристаллическом экране мы видим, что температура стала больше 40 градусов.

На слайде мы видим, что сейчас, когда датчик влажности почвы помещен во влажную почву, он показывает значение 583 единицы, что соответствует влажности грунта $> 8\%$ и полив не требуется.

Датчик влажности почвы я помещаю в сухую землю. В программе мы используем крайние значения: если земля сухая и требуется полив, то показания будут от 650 до 1200 .

Мы видим на дисплее, что влажность почвы составляет 943 , что соответствует значениям $< 8\%$, поэтому включается насос и происходит полив. Когда значение уменьшается до < 650 , то насос выключается (слайд).

Работа над данным макетом продолжается, мы планируем включить сюда еще дополнительные элементы и доработать программу.

Так, мы сделаем полный полив, частичный или разбрызгивание в зависимости от показаний датчика влажности почвы.

Доделаем систему энергообеспечения с помощью накапливания ее в одном мощном аккумуляторе; инфракрасный или простой обогреватель будет запитываться от аккумулятора, который заряжается от солнечной батареи. Солнечную батарею мы хотим разместить рядом с теплицей, чтобы управлять ее движением вслед за солнцем.

Демонстрация образца родителям и родственникам выявила огромный интерес к этому комплекту. Все, без исключения, отметили, что это будет просто прекрасное устройство в помощь огороднику.

Таким образом, я смог подтвердить свою гипотезу: «умная теплица», изготовленная своими руками, может применяться на любом приусадебном или мичуринском участке.

Используя проектно-исследовательский метод, я создал собственный продукт.

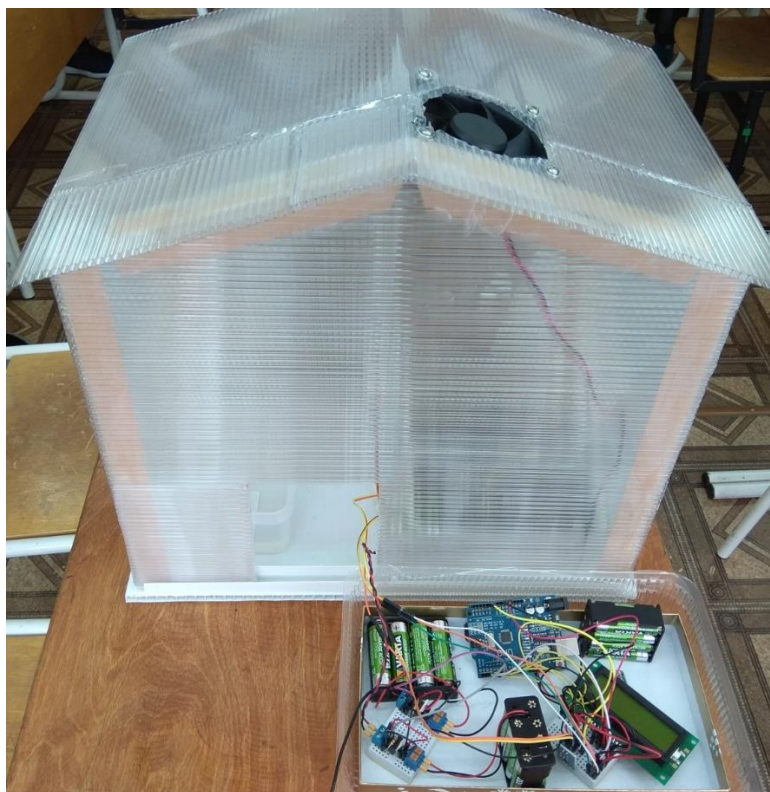


Рисунок 1 –Макет «умной теплицы» «Кормилицы».

ДЕГИДРАТОР ФИЛАМЕНТА ДЛЯ 3D ПРИНТЕРА

Кузнецов Даниил

МОУ Лицей №1, 10 класс

Руководитель: Попов Михаил Викторович, учитель информатики

Введение

Актуальность. В настоящее время услуги по использованию 3D принтера актуальны, как никогда ранее. Если раньше 3D принтеры использовались только в больших компаниях и были недоступны для покупки в частные руки, то сейчас же большое количество людей приобретает его для личного пользования.

Подготовка и формулировка проблемы. В современном мире уже невозможно обойтись без печати 3D моделей, так как человечество развивается, таким образом облегчая себе жизнь, но проблема выявляется в том, что принтеры усложняют жизнь своей некорректной печатью из-за

гидроскопичности пластика. Для таких случаев эффективными являются удаление влаги из пластика. Поэтому нам пришла идея собрать собственный дегидратор феломенты на основе Arduino Nano, с помощью которого можно будет подготовить пластик к печати.

Цель работы: собрать дегидратор феломенты для 3D принтера.

Методы исследования: аналитический, информационное моделирование, компьютерное моделирование, метод обработки статистических данных.

Разработанность проблемы: В современном мире стали часто пользоваться 3D принтерами, ведь благодаря ему можно легко сконструировать любые модели, что значительно облегчает жизнь современных людей. Но возникают проблемы с эксплуатацией 3D принтера, весь пластик гигроскопичен, и поэтому собирает влагу из воздуха, из-за чего получаемые детали в итоге плохого качества, проявляются дефекты из-за наличия воды в пластике, которая в процессе печати испаряется, образуя пузыри и пустоты в пластике, что недопустимо для важных компонентов. Для таких случаев эффективными являются сушка феломенты. Поэтому мы решили собрать собственный дегидратор феломенты на основе Arduino Nano, которая максимально избавит наш пластик от влаги.

Разработанность исследуемой проблемы. На сегодняшний день мы сравнили и прочитали все про дегидраторы в магазинах, проблема устройства, это его цена, поэтому мы решили собрать не дорогой аппарат своими руками. Если владелец 3D принтера сталкивается с проблемой несхожестю детали с оригиналом, он задумывается о том, как усовершенствовать принтер своими руками, думая, что проблема в нем самом. Оказывается, решить эту проблему и сделать дегидратор не такой уж и сложный процесс – важно хоть немного разбираться в электронике и программировании, и смастерить дегидратор не составит большого труда. Наш проект удобен и прост в использовании, в любой момент можно легко изменить программу и модернизировать, можно добавить в программу большее количество видов пластика.

Основная часть

Гипотеза: с помощью Arduino Nano можно создать дегидратор феломенты для 3D печати.

Цель: собрать прототип системы сушки 3D пластика.

Задачи:

1. Узнать температуру размягчения всех используемых видов пластика.

2. Спроектировать собственную систему дегидратора на основе Arduino,
3. Найти необходимые компоненты для конструирования.
4. Используя доступные и бесплатные онлайн сервисы, такие как (vectr.com и tinkercad.com), смоделировать и изготовить при помощи 3D принтера и лазерного станка необходимые компоненты для создания корпуса для экрана.
5. Написать программу для функционирования в среде Arduino IDE.
6. Провести испытания прототипа на разных типах пластика.

Методы исследования: аналитический, информационное моделирование, компьютерное моделирование.

Этапы выполнения работы.

Первый этап – спроектировать собственную систему сушки пластика (Рисунок 1).

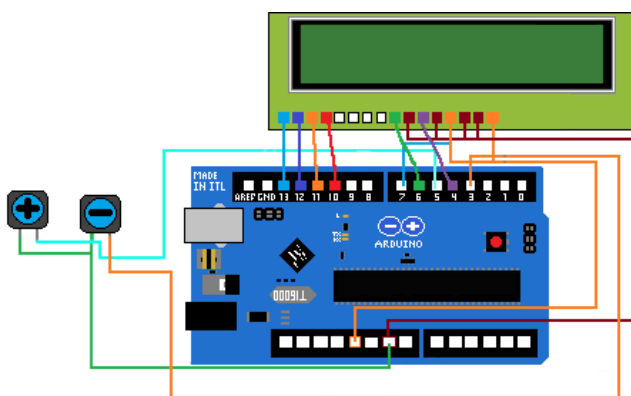


Рисунок 1 –Макет системы сушки пластика

Второй этап - подобрали необходимые компоненты для конструирования.

Таблица 1 – Необходимые компоненты

		<p>Кнопки</p>	<p>14 руб</p>

		LCD экран	150 руб
		Arduino nano	50 руб

Третий этап - используя сервис Vectr Labs, нарисовали компоненты.

Четвертый этап- сделали корпус для экрана .

В результате получился корпус для экрана (Рисунок 2).



Рисунок 2 –Корпус экрана

Для распечатки компонентов корпуса использовали PLA пластик. Каждая деталь печаталась на 3D принтере. В результате получился корпус (Рисунок 4).

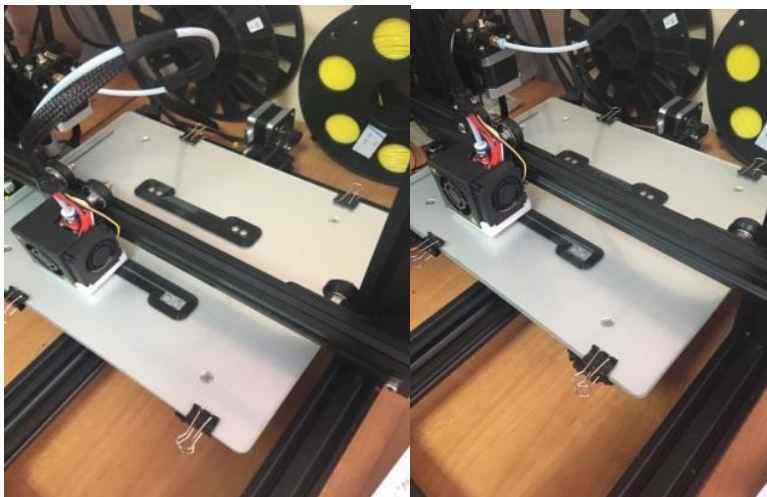


Рисунок 3 –Процесс изготовления корпуса

Вывод: распечатали и собрали корпус.

Шестой этап - используя приложение Arduino IDE написали программу для функционирования дегидратора.

Вывод: написали программу для функционирования дегидратора.

Седьмой этап - собрали дегидратор и провели испытания прототипа на различных типах пластика



Рисунок 4 –Корпус

Вывод: провели испытания дегидрататора на различных видах пластика. Он удобен и прост в использовании, в любой момент можно добавить в программу свой вид пластика.

Заключение

Спроектирован собственный дегидрататор феломента 3D принтера. Нашли и купили компоненты для конструирования проекта. Подобрали необходимые компоненты для создания модели корпуса для экрана.

Написали программу для функционирования температуры пластика.

Провели испытания дегидрататора на пластике.

В результате мы собрали дегидрататор для максимальной сушки пластика

Наш проект удобен и прост в использовании, в любой момент можно добавить свой вид пластика.

Таким образом, мы подтвердили выдвинутую нами гипотезу и достигли поставленные цель и задачи.

МЕТОДЫ ДЕТЕКТИРОВАНИЯ ЗВУКОВЫХ СИГНАЛОВ НА ДОРОГАХ ОБЩЕГО ПОЛЬЗОВАНИЯ

Основская Юлия

*Муниципальное бюджетное образовательное учреждение лицей при ТПУ,
10 класс, г. Томск*

Руководитель: Воротов Вячеслав Евгеньевич

Беспилотные автомобили постепенно набирают популярность. Но на текущий момент в них никак не распознается звук. Существующие сейчас модели используют ультразвуковые датчики, радары, камеры и лазеры для сбора информации о дороге. Их автопилот полагается на детальные карты, создаваемые с помощью этих сенсоров, и методы детектирования объектов, но не звука. Тем не менее, звук является важным фактором на дороге, так как существуют необходимые сигналы, которые водитель или автопилот должен распознавать. Например, важно слышать сирены специальных служб, гудки других машин, звук удара и так далее.

Нейросеть должна ускорять процесс принятия решений в случае обнаружения звукового сигнала, предвещающего столкновение. Иными словами, “услышав” глухой резкий звук или звук гудка, нейросеть должна уменьшить интервал снятия показаний с сенсоров. Зафиксировав звук

сирены, автомобиль должен уступить дорогу. Все эти решения нейросеть должна принимать на основе подающихся ей данных, поэтому важно уметь отличать важные сигналы от ненужных шумов, присутствующих на дороге.

Существуют алгоритмы распознавания звука, способы получения незашумлённой звуковой дорожки.

Для избавления от ненужных шумов предлагается использовать метод подавления шума. Звуковая дорожка переводится в спектрограмму - зависимость спектральной плотности мощности сигнала по времени. Представлять спектрограмму можно как двумерную диаграмму, где на горизонтальной оси отложены временные отсчеты, а на вертикальной - значения частоты сигнала. Тогда следующими шагами будет:

- Провести преобразования Фурье над исходным сигналом.
- Провести анализ построенной спектрограммы.
- Перемножить полученную маску со спектром.
- Выполнить обратное преобразование для получения сигнала. [1]

На дорогах звуки часто перекрывают друг друга. Человек способен выделить из гама определенные звуки. Нейросети для определения слышимых одновременно звуков необходимо иметь множество сделанных человеком аннотаций. Поэтому более целесообразно обучать нейросеть на выборке индивидуальных звуков и тестировать на полифонических записях. [2] предлагает использовать Метод Главных Компонент для определения количества источников звуков, Неотрицательное Матричное Разложение для их дальнейшего отделения и Скрытую Марковскую Модель для их распознавания.

При отдельном определении звука и его источника может возникнуть проблема ассоциации распознанного звука и направления его прибытия. Для решения этой проблемы [3] предлагает распознавать звук и источник одновременно.

[4] предлагает способ определения резких звуков. Частотно-временная карта, полученная с помощью преобразования Фурье, объединяется со звуковыми функциями, полученными как из карты так и из оригинального звука, с помощью матрицы перехода. Полученные данные отдаются Сверточной Нейронной Сети.

Используя представленные алгоритмы, которые необходимо объединить и настроить под задачи работы, нейросеть должна уметь

правильно определять источник звука - правильно расставлять коэффициенты соответствующие различным звуковым сигналам.

При этом важно понимать, что сигналы, которые должна распознавать сеть (сирена, гудок и звук удара) не уникальны. К примеру, звук полицейских сирен может быть вставлен в музыку. Отличать такие вставки от действительных сирен поможет длительность сигнала и уменьшение или увеличение длины звуковых волн. Как правило, сирена специальных служб слышна длительное время и либо удаляется, либо приближается к автомобилю.

Так как беспилотные автомобили - сложные системы, микрофоны и наша нейросеть будут работать в команде с остальными сенсорами машины. Поэтому при обнаружении звука, искусственный интеллект автомобиля может обратиться к остальным сенсорам для подтверждения полученного сигнала.

Выводы

Были проанализированы имеющиеся источники, оценена работоспособность существующих беспилотных автомобилей на предмет обработки звуковых сигналов окружения, подобраны оптимальные методы решения задачи.

На сегодняшний день не существует аналогов представленной системы, в то время как распознавание звуков важно на дорогах.

Область беспилотных автомобилей не стоит на месте. В скором времени мои предложения будут реализованы и протестированы.

СПИСОК ЛИТЕРАТУРЫ:

1. Метод обнаружения посторонних звуков с применением нейронных сетей / С. Д. Гречков, Н. В. Размочаева, Д. М. Клионский, ННБVIII, Санкт-Петербург, 14 –16 мая 2020
2. Sound Event Detection using a Neural Network / Aditya Agarwal, Syed Munawwar Quadri, Savitha Murthy, Dinkar Sitaram / ICACCI September 2016
3. Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks / Sharath Adavanne, Archontis Politis, Joonas Nikunen and Tuomas Virtanen arXiv:1807.00129v3 [cs.SD] 17 Dec 2018
4. Anomalous Sound Detection Based on Convolutional Neural Network and Mixed Features / Jie Zhao 2020 J. Phys.: Conf. Ser.1621 012025

«РОБО - ПАУК»

Подымов Николай

*Муниципальное автономное общеобразовательное учреждение
средняя общеобразовательная школа №2 с. Александровское*

Александровского района Томской области

8 класс, с. Александровское

Руководитель: Баротов Н. М, учитель физики и робототехники

Аннотация

Мир роботов имеет огромное значение в жизни человечества. Каждый день появляются все новые и новые технологии которые помогают, облегчают нам обыденную жизнь. Изучении робототехники вызвала во мне интерес уже со времен начальной школы. В этой работе я собрал результаты моих долгих учений. Я собрал робота паука из набора Lego Mindstorms ev3. Запрограммировал робота на языке программирования LEGO MINDSTORMS Education EV3. Мой проект робо - паук практически ничем не отличается от обычного паука. Он умеет замечать препятствий которые находятся напротив него и убегать от них. При сборке данного робота я использовал 2 больших моторов и 1 ультразвуковой датчик. Мой робо - паук имеет 4 механических ног.

Ключевые слова

Робо – паук, большой мотор, ультразвуковой датчик, язык программирования, программа, механизм, шестеренка, балка и т. д.

Введение

Наука не стоит на месте. Уже сейчас уровень развития робототехники достиг больших высот. Писатели-фантасты неоднократно пугали мир разнообразными вариациями на тему «бунта машин». Но ситуация с развитием робототехники в настоящее время складывается таким образом, что остановить это развитие в данной сфере невозможно. А все потому, что роботы уже заняли свою нишу в жизни общества. Они стали частью современной промышленной революции, характеризующейся широким внедрением адаптивных технологий и роботизацией производства. Ежегодно все больше предприятий автоматизируется, поэтому на данный момент завод, на котором работает всего несколько десятков человек, а всю основную работу выполняют роботы, уже никого не удивляет. Промышленные роботы выпускаются десятками тысяч. Несмотря на то, что

этот рынок достаточно давно сформировался, с выходом на него Китая ситуация здесь только обостряется. {1}

Сегодня нам сложно представить свою жизнь без роботизированных технологий. Нас везде окружают роботизированные механизмы. Конечно, это хорошо, ведь без роботов наша жизнь была бы, намного, сложно, чем можно представить себе. Я изучаю мир роботов уже несколько лет. В нашей школе есть почти все необходимое для полноценного развития в технологическом плане. На данном проекте я собрал роботизированного автономного паука. Он ничем не отличается по поведению от обычного паука. Конечно, он пока не умеет лазать по стенам и ползать по деревьям. Но он умеет определять препятствий у себя в пути, смотреть по сторонам и автоматически определять траекторию дороги где нет препятствий и двигаться в ту сторону. В дальнейшем мою работу можно будет применять в турнирах по гонкам по линии для шагающих роботов.

При сборке робота использовались два больших мотора, которые отвечают за шаги 6 механических ног. Так же 1 ультразвуковой датчик, который отвечает за определении безопасной траектории.



Актуализация

Актуальность данной работы связана с предметной частью исследования. Робототехника является важнейшим аспектом в современной жизни человечества. Мы представить не можем свою жизнь без современных роботизированных технологий. Развивая робот технологию, мы развиваем свою жизнь, свое реальность. Машины были и будут актуальным в нашей жизни. Мой проект можно использовать как в образовательных, так и в исследовательских целях. {2}

Цель:

- Собрать робота, который не имеет колеса, но что б он передвигался.
- Написать программу на языке LEGO MINDSTORMS Education EV3.
- Сделать робота автономным.

Задачи:

- Изучить основы робототехники, набор Lego Mindstorms
- Изучить принципы работы датчиков, моторов Lego
- Научиться собирать конструкции
- Научиться написать программу для робота на языке LEGO MINDSTORMS Education EV3.

Гипотеза: Собрать робота который не будет иметь колеса, но что он смог двигаться. Так же, что б он умел определять безопасную траекторию, и что б все это было автономным.

Основная часть

1. Процесс сборки

Процесс сборки проекта начался в декабре месяца, сразу после турнира кубка губернатора томской области по робототехнике, где мою работу признали самым техничным. Сама процес сборки практически не вызвало во мне сложности.

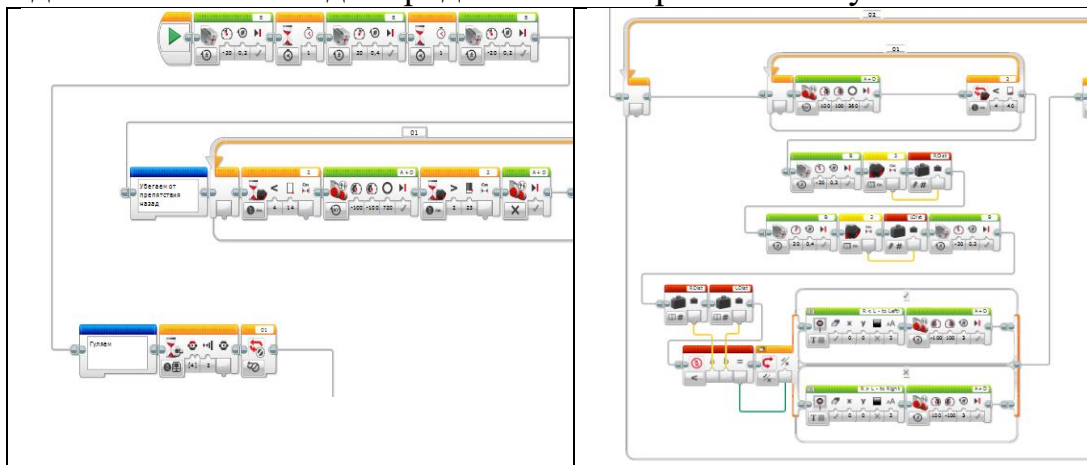


Самой сложной частью при сборке стало конструкция ног паука. На весь процесс сборки было потрачено около 3 недель времени. При сборке использовал 2 больших моторов, 1 средний мотор, 1 ультразвуковой датчик.

2. Программное обеспечение

Написании программы стала самой сложной частью проекта. На написании программы было потрачено около 4-5 недель. При написании программы мне помогли мой руководитель и старшеклассники. Алгоритм программы выглядит следующим образом:

- 1) После включения средний мотор на которой прикреплен ультразвуковой датчик поворачивается на право на 70 градусов, ждет 1 секунду.
- 2) За это время ультразвуковой датчик рассчитывает дистанцию до ближайшего препятствия.
- 3) Затем средний мотор поворачивается на лево на 140 градусов и ждет 1 секунду
- 4) За это время ультразвуковой датчик рассчитывает дистанцию до ближайшего препятствия.
- 5) После блок управления выполнить математический расчет и выберет ту траекторию, в которой свободная дистанция больше
- 6) Робот начинает, вращаясь через пункт рулевого управление двигаться в ту сторону, пока пустая дистанция не закончится.
- 7) Когда робот доедет до конца пустой траектории робот повторить пункт 1-4
- 8) Сделает 3 шага назад и продолжить алгоритм 5 и 6 пункта.



Заключение

Работа над проектом была долгая, но очень интересная. Я считаю, что достиг поставленной цели проекта. робот собран, она автономная. В ходе выполнения данного проекта я познакомилась с многими вещами которые является для меня новинкой. Много чему научился.

Дальнейшие планы

В дальнейшем я буду более интенсивно изучать основы написания программы на языке LEGO MINDSTORMA EDUCATOINS EV3. Так же хочу проверить способности данного робота на соревнования по робототехники. А именно в турнире Шагающих роботов и в турнире прохождении лабиринта.

СПИСОК ЛИТЕРАТУРЫ.

1. Официальный сайт РОБОВИКИ. Ссылка доступа: <https://robo-wiki.ru> (дата обращения 15.01.2021)
2. Официальный сайт компании Lego. Ссылка доступа: <https://www.lego.com/ru-ru/categories/robotics> (дата обращения 19.01.2021)
3. Интернет сайт PRO ROBOT. Ссылка доступа: <https://www.prorobot.ru/lego.php> (дата обращения 20.01.2021)

STTM – МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ОБМЕНА СООБЩЕНИЯМИ

Приставка Егора Алексеевича

*Муниципальное бюджетное общеобразовательное учреждение
города Иркутска лицей №3, 11 класс*

Руководитель: Вайншенкер Анна Борисовна

Введение

Мессенджер — это программа (приложение) для смартфона или персонального компьютера, позволяющая **мгновенно обмениваться** с текстовыми сообщениями с другими пользователями . Кроме того, мессенджер может включать с себя такие функции как: аудио- и видеозвонки, передача видео- , аудио файлов, изображений и аудио сообщений.

Наличие функции голосовых сообщений позволяет упростить отправку сообщений – этот способ не требует нажатия большого числа кнопок: достаточно зажать одну кнопку и говорить в микрофон. Однако такие сообщения не всегда удобны для восприятия собеседником: во-первых, для их прослушивания необходим динамик или наушники и в

случае отсутствия последней информация, содержащаяся в сообщении может стать известной окружающим собеседника людям. Во-вторых, длина голосовых сообщений может достигать тридцати секунд, и у собеседника не всегда есть возможность прослушать столь длинное сообщение.

В связи с вышеописанными недостатками была поставлена **цель**: создать мессенджер для ОС Android с функцией перевода голосовых сообщений в текст.

Были поставлены следующие задачи:

- 1) Изучить принцип работы клиент-серверных приложений
- 2) Изучить особенности приложений для операционной системы Android
- 3) Создать сервер на ЯП Python
- 4) Создать функции перевода голосовых сообщений в текстовые, используя ресурсы сервера и библиотеку SpeechRecognition
- 5) Создать приложение-клиент для ОС Android, обладающее функцией перевода голосовых сообщений в текстовые и использующее функции для распознавания речи, созданные ранее.
- 6) Провести закрытое альфа-тестирование приложения, используя группу в социальной сети Вконтакте с целью выявления всех недостатков приложения
- 7) Опубликовать исходный код приложения (как клиента, так и сервера) на платформе GitHub с целью дать программистом со всего мира использовать код данного приложения в собственных проектах.

Создание сервера

В данном проекте сервер реализован на языке программирования Python. Данный язык программирования был выбран по следующим причинам

- 1) Простота синтаксиса
- 2) Простота создания серверных приложений, используя классы «HTTPServer» и «BaseHTTPRequestHandler»
- 3) Высокая скорость работы приложения, разработанных на данном языке программирования.

Функционал сервера реализует класс написанный вручную класс JSONRequestHandler. Данный класс описывает методы, при помощи которых приложение-клиент может отправить запрос на сервер, а так же выполняющий необходимые функции(отправка сообщения пользователю, удаление аккаунта, процедуры авторизации и регистрации и другие

функции), в зависимости от запроса. В качестве примера приведена обработка запроса-регистрации.

При получении запроса, сервер проводит проверку, содержит ли запрос поле «action» (действие) и в зависимости от результата проверки, либо выполняет запрос, либо отправляет клиенту сообщение об ошибке следующего содержания.

Создание приложения клиента

Активности (вкладки) авторизации и регистрации имеют общую структуру: они состоят из трёх текстовых полей, в которые необходимо ввести имя пользователя, пароль, а так же дополнительный пароль (PIN-Код, состоящий из ряда цифр) как дополнительная степень защиты. На обеих вкладках есть две кнопки, одна из которых отправляет пользователя на вкладку регистрации (в случае, если у пользователя нет аккаунта) или на активность авторизации(если у пользователя есть аккаунт) (*приложение 4*).

Активность «Создание аватара»

Данная активность предназначена для создания изображения своего профиля (так называемого «Аватара»). Она имеет следующую структуру: 3 кнопки: «загрузить изображение с устройства», «продолжить», использовать стандартный.

При нажатии кнопки «загрузить изображение с устройства» открывается галерея, и пользователь имеет возможность выбрать изображение, которое будет использоваться в качестве «аватара». При выборе изображения, происходит процесс перекодирования изображения в формат Base64 (стандарт кодирования двоичных данных при помощи только 64 символов ASCII). В последствии, при нажатии кнопки «продолжить» клиент отправляет запрос, на сервер, и после получения ответа переходит на активность «Центр сообщений» . Аналогичное происходит при нажатии кнопки «использовать стандартный», однако вместо данных изображение на сервер отправляется строка «nil»

Активность «Центр сообщений»

Активность «Центр сообщений» предназначена для выбора пользователем собеседника из списка друзей. Активность имеет три кнопки: «Помощь», нажав на которую пользователь будет перемещён на активность, объясняющую принципы использования данного приложения, кнопку «найти друзей», после нажатия на которую пользователь будет отправлен

на активность «найти друзей», а также кнопку «выход», при нажатии на которую откроется меню настроек приложения.



Рисунок 1 –Центр сообщений

Активность «Переписка»

Активность состоит из нескольких элементов: одного RecyclerView, трёх кнопок: «FL» (сокр.от Fille «файл»), «IMG» (сокр. От Image - изображение), и «отпр» (отправить), а также поля для ввода сообщения.

При старте активности клиент начинает отправлять запрос на получение сообщений каждые 4 секунды. Полученными данными через адаптер заполняется RecyclerView. При нажатии кнопок FL и IMG пользователь должен прикрепить файл или изображение соответственно, после чего происходит отправка данных файлов, а также введённого текста собеседнику. Собеседнику нужно нажать по пришедшему сообщению, чтобы сохранить файл.

Активность «Поиск друзей»

Чтобы найти пользователя, необходимо ввести данные в поле для ввода и нажать кнопку «найти». После выполнения этих действий, клиент отправляет запрос и получает 2 массива: один содержит имена пользователей, второй – их аватары. Затем происходит заполнение данными RecyclerView и их визуализация. Чтобы добавить человека в друзья, необходимо просто нажать на нужного пользователя, после чего он появится в вашем списке друзей и можно будет начать переписку.

Активность «Помощь»

Активность предназначена для новых пользователей, не знакомых с функционалом приложения. На ней представлены изображения всех элементов приложения, а также инструкция по работе с ними.

Проведение закрытого альфа-тестирования

Тестирование данного проекта происходило через закрытую группу в социальной сети Вконтакте. В ней был размещён установочный файл приложения, функционал приложения, а так же время, в которое его можно было использовать, ввиду отсутствия возможности держать сервер включенным постоянно.

Всего в закрытом альфа-тестировании приняло участие 50 человек со всей России, которые, в последствие, оставили свои отзывы о приложении.

Тестирование выявило следующие недостатки приложения

- 1) Прекращение работы приложения при передаче фотографий больших размеров (от 4 Мб)
- 2) Необходимость введения подтверждения запроса на добавление в друзья
- 3) Появление с небольшим шансом «фантомных» сообщений при одновременной передаче голосового сообщения и фотографии.

Заключение

Операционная система Android очень популярна. Получив свою популярность ещё в 2012 году, “зелёный робот” до сих пор активно используется. Приложение для обмена сообщениями для этой ОС – одни из самых популярных приложений в мире, они установлены почти на всех андроид-устройствах.

Все поставленные задачи были выполнены: приложение создано, тестирование, выявившее недостатки приложения, которые будут устранены в следующем обновлении, проведено, а исходный код приложения опубликован на платформе GitHub.

СПИСОК ЛИТЕРАТУРЫ:

- 1) Habr [электронный ресурс]: русскоязычный веб-сайт в формате коллективного блога. Режим доступа: <https://habr.com/ru/post/109944/>, дата обращения: 25.09.2020.
- 2) Docs.Oracle [электронный ресурс]: техническая документация на продукцию компании Oracle. Режим доступа: <https://docs.oracle.com/javase/7/docs/api/>. Дата обращения: 11.02.2020

- 3) Сайт Александра Климова [электронный ресурс]: уроки по разработке Android приложений/ А.Климов. Режим доступа: developer.alexanderklimov.ru/android/. Дата обращения: 03.02.2020
- 4) Г.Шилдт«Java 8. Руководство для начинающих». [Текст]/. Г.Шилдт, Киев: Диалектика, 2015, 320 с.
- 5) Блох Д. «Эффективное программирование.» [Текст] / Блох Д – М: Лори, 2014, 310 с
- 6) David Beazley, Brian, Jones, Brian К « Python Cookbook » [Текст] / Rubens Prates, 2019, 649
- 7) Кей С. Хорстманн, Гари Корнелл «Java. Библиотека профессионала.» [Текст] / Кей С. Хорстманн, Гари Корнелл – М: Вильямс, 2014, 430 с.
- 8) Приставка Е.А, «Создание собственного приложения для операционной системы Android», 2019, С 3-8

КОМПЬЮТЕРНЫЙ ПОДХОД К РЕШЕНИЮ И ГЕНЕРАЦИИ ШАШЕЧНЫХ ЗАДАЧ

Сокольников Михаил Юрьевич

МБУДО г. Иркутска ЦДТТ, МАОУ «Лицей ИГУ» г. Иркутска, 9 класс

Руководители:

Рейнгольд Григорий Борисович, педагог дополнительного образования
в.к.к. МБУДО г. Иркутска ЦДТТ;

Рудых Александр Николаевич. педагог дополнительного образования
МБУДО г. Иркутска ЦДТТ

Шашки, хотя и находятся в тени шахмат, представляют несомненный интерес, как с точки зрения математики, так и с точки зрения программирования. Несмотря на свою кажущуюся простоту, по сравнению с шахматами, они имеют гораздо большую математическую составляющую. Например, благодаря однотипности фигур имеет смысл игра на поле произвольных, четных, размеров.

Автор данной работы, несмотря на многочисленные и успешные компьютерные реализации этой игры, решил попытаться счастья на этом поприще. На данном этапе было решено не создавать автоматического игрока, а начать с автоматического решения задач, на основе которого, в будущем создать полноценный генератор шашечных задач.

Цель работы: создать компьютерную программу для решения шашечных задач, на основе которой, после, создать и генератор шашечных задач.

Задачи:

- Произвести анализ аналогичных разработок.
- Сделать программу «демонстрационную доску», реализующую соблюдение всех правил при игре человек-человек.
- Сделать программу «решатель задач».
- На основе «решателя» создать «генератор задач»

Программа реализована на языке Python. Надо отметить, что реализована игра на поле произвольных размеров. При этом за основу были взяты правила русских шашек (с учетом произвольности размера поля) [1].

Актуальность работы: отсутствие реализаций решений и/или генерации шашечных задач при наличии подобных программ для игры «Шахматы»

Библиографический обзор

Игра в шашки известна со средних веков. Точное время и место создания этой игры неизвестны [1]. Существует несколько видов этой игры, которые различаются, как размерами игрового поля, так и некоторыми мелкими деталями правил [2], [3].

Разумеется, в последние десятилетия большинство интеллектуальных игр, не осталось без внимания программистов, и были созданы компьютерные программы, играющие на равных, а то и сильнее с лучшими игроками-людьми. И шашки не составляют исключение [4], [5], [6]. Поэтому, как сказано выше, было принято решение пока не делать автоматического игрока.

Что касается автоматического решения задач, то в этом направлении так же сделано очень много [7], [8]. Но автоматическая генерация нигде не обнаружена.

Элементы оригинальности заключается в автоматической генерации задач, в доске произвольных размеров, в сочетании в одной работе, как прямой игры, так и «поддавок».

Но заметим, что алгоритмов и кодов, по которым реализованы найденные разработки, найти не удалось, и всё делать самостоятельно.

Демонстрационная доска

В начале работы была сделана демонстрационная доска, то есть программа, в которой могут играть 2 человека. При этом невозможно делать ходы с нарушением правил.

Шашечная доска представлена в программе как двумерный массив размером $n*n$. Пустые клетки представлены 0, белые шашки – 1, чёрные шашки – 2, белые дамки – 3, чёрные дамки – 4.

Каждое поле обозначено индексами соответствующего элемента массива. Ход задаётся полем, где находится фигура и полем, куда её следует поставить. В случае хода, с многократным взятием, задаются все промежуточные поля. Сперва проверяется возможность хода, и в случае возможности он делается.

В результате хода появляется новое положение. Убираются взятые фигуры, проверяется достигнута ли шашкой противоположная горизонталь, если это случается тип фигуры меняется с шашки на дамку (в случае многократного взятия, это может случиться в середине хода). Передаётся очерёдность хода.

После каждого хода проверяется, достигнут ли выигрыш?

В случае попытки сделать неправильный ход (нарушение очерёдности, уклонение от взятия, ход в невозможную сторону, либо на невозможное расстояние и прочее), положение не меняется и ход не передаётся.

Есть две возможности задания хода: с помощью написания адресов соответствующих клеток вручную и с помощью графического интерфейса.

Решение одноходовок

Далее, был реализован алгоритм решения задач «выигрыш в один ход», который проводит полный перебор всех возможных ходов.

Это делается следующим образом:

- Составляется список всех возможных ходов.
- Далее эти ходы последовательно перебираются, и полученное положение анализируется на возможный выигрыш. Если найден хотя бы один вариант, выдаётся сообщение об этом.

Решение многоходовок

При решении задач «Выигрыш в несколько ходов» использован предыдущий алгоритм. Это происходит следующим образом:

- Составляется список всех возможных ходов из начального положения.

- Эти ходы перебираются, и для каждого получаются списки следующих положений. Если получается выигрыш, то работа по дальнейшему перебору в этом направлении останавливается и производится переход на следующий свой ход.
- Составляется список всех возможных ответных ходов, и получившиеся в результате них положения передаются в пункт 1.
- Когда сделан весь перебор на заданную глубину, смотрится, на все ли ответные ходы противника есть выигрышные ходы, и если находится хотя бы один такой ход (на который нет выигрышного продолжения), то ответ отрицательный.
- В конце выдаётся количество найденных решений и сами решения.
- Данная часть работы закончена и успешно прошла испытания. В настоящее время уверенно решаются задачи на выигрыш за произвольное количество ходов.

Генерация задач

При генерации задач используется алгоритм для решения задач «Выигрыш в несколько ходов». Это происходит следующим образом:

- Как основа для будущей задачи берется начальное положение для текущего размера поля
- Составляется список всех возможных ходов из текущего положения для ходящей стороны
- Совершается один, случайный ход, после чего ход передается следующей стороне
- После каждого хода с помощью алгоритма «решателя» проверятся, возможно ли текущую позицию решить за определенное количество ходов (количество ходов задается заранее), если это так, то на демонстрационной доске отображается эта позиция и на экран выводятся все последовательности ходов, которые были получены в ходе работы «решателя», иначе перебор продолжается и совершается еще один ход.
- Если в ходе работы «решателя» у ходящей стороны закончились возможные ходы, а задача еще не получена, происходит возврат к изначальной позиции и цикл повторяется

Заключение

- Задачи, поставленные на первом этапе, в целом, выполнены. Планируется продолжить работу по следующим направлениям:
- Проверка решателя задач на разных позициях.
- Усовершенствование случайного генератора задач.
- Добавление возможности выбора сложности генерируемой задачи
- Реализация решения и генерации задач не только на выигрыш, но и на наилучший ход
- Проведение исследования с целью выяснить, какой из игроков имеет преимущество в начальной позиции на досках разных размеров. Исследование имеет смысл, поскольку на доске размеров 4*4 при обоюдной правильной игре всегда будут выигрывать чёрные.
- Провести дополнительные исследования с вариацией игры в «поддавки»
- Попробовать решить вопрос, существуют ли такие положения, когда неоспоримое преимущество у одного из игроков имеется как в обычной игре, так и в поддавках.

СПИСОК ЛИТЕРАТУРЫ:

1. Шашки [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Шашки>
2. Герцензон Б., Напреенков А. – «Шашки — это интересно. Учебник шашечной игры», СПб.: Издательство "Литература", 1997
3. В.Б. Городецкий, «Книга о шашках», М. «Детская литература», 1984
4. Компьютерные новости: компьютерные шашки [Электронный ресурс]: Шашки в России – Режим доступа: <http://www.shashki.com/Article369.html>
5. Шашечные программы [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/Категория:Шашечные_программы
6. Компьютер против Человека в шашках. Есть ли интрига? [Электронный ресурс]: Livejournal – Режим доступа: <https://anikeev-yuriy.livejournal.com/117512.html>
7. Программа для решения шашечных задач в режиме онлайн [Электронный ресурс]: ФЕНИКС-64 шахматно-шашечный web-клуб – Режим доступа: <http://fenix64.com/programma-dlya-resheniya-shashechnyx-zadach-v-rezhime-onlajn/>

8. Обзор современных компьютерных программ по русским шашкам [Электронный ресурс]: ТАВЛЕИ, Компьютерные шашки– Режим доступа: https://plus.gambler.ru/tavlei/comp/comp_1.htm

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ПОКУПКИ КНИГ

Ушаков Фёдор Андреевич, Волжин Никита Вячеславович

*Муниципальное бюджетное общеобразовательное учреждение лицей
при ТПУ г. Томска, 10 класс*

Руководитель: Буркатовская Юлия Борисовна, учитель информатики

Рекомендательные системы – системы машинного обучения, помогающие пользователям находить новые продукты и службы. Каждый раз, когда вы покупаете что-либо онлайн, рекомендательная система направляет вас к тому товару, который вам скорее всего понравится.

Рекомендательные системы крайне важны в современном виртуальном мире, т.к. пользователям часто нужна помощь в нахождении того, что им нужно. Правильное применение рекомендательных систем приводит к большему удовлетворению пользователей и, конечно, большему количеству продаж. В наше время они настолько обыденны, что ими можно пользоваться, даже об этом не подозревая. Когда мы просто физически не можем просмотреть весь контент, имеющийся на сайте, рекомендательные системы помогают нам. Конечно, по причине их важности можно увидеть большое кол-во статей, так или иначе связанных с рекомендательными системами[1-3].

Использование рекомендательных систем можно увидеть на «AliExpress», «КиноПоиске», в приложении «YouTube», в составлении радио на «Я.Музыке», даже в ленте популярной социальной сети «ВКонтакте».

Тем не менее, не все из существующих систем совершенны. Авторы работы столкнулись с тем, что при поиске книг на сайте интернет-магазина «Лабиринт», рекомендуемые товары не соответствовали интересам. Это вызвало желание создать альтернативную рекомендательную систему.

Важнейшей целью работы стала разработка рекомендательной системы, которая тщательно рассматривает интересы пользователя.

Задачи работы:

1. Выяснить, имеется ли проблема несоответствия рекомендаций интересам у других пользователей.
2. Изучить виды рекомендательных систем и алгоритмы, используемые в них.
3. Создать базу данных с различными пользователями и их интересами в литературном сегменте.
4. Создать рекомендательную систему, удовлетворяющую пользователей.

Для обоснования актуальности проблемы было проведён опрос среди обучающихся МБОУ ЛТПУ г. Томска. Ученикам предлагалось оценить, насколько часто рекомендуемый им контент на сайтах интернет-магазинов книг соответствовал их интересам по десятибалльной шкале. По результатам опроса было выяснено, что лишь 25% поставили оценки, выше 5, а 58.3% поставили оценки ниже 5. Таким образом, было установлено, что проблема является распространённой. Более подробная статистика представлена в таблице ниже.

Таблица 1 – Результаты опроса

1	2	3	4	5	6	7	8	9	10
0%	8.3%	16.7%	33.3%	16.7%	8.3%	0%	8.3%	8.3%	0%

После проведения опроса была выбрана целевая аудитория проекта: интересующиеся чтением подростки в возрасте от 15 до 18 лет.

Описание принципа работы и видов рекомендательных систем[1,5]

В основе работы рекомендательных систем лежит установление связей. Могут быть установлены связи вида пользователь-товар, товар-товар и пользователь-пользователь.

1. Связь вида пользователь-товар

Данный вид связи возникает, когда у пользователя имеются предпочтения касательно определённого вида товаров. Например, спортсмену понадобятся товары, связанные со спортом, поэтому рекомендательная система установит связь «спортсмен-спортивный инвентарь».

2. Связь вида товар-товар

Данный вид связи возникает, когда предметы похожи между собой (как по внешнему виду, так и по описанию).

3. Связь вида пользователь-пользователь

Данный вид связи возникает, когда пользователи имеют схожие вкусы и отдают предпочтения одинаковым видам товаров.

Устанавливая связи, рекомендательная система может предсказать оценку пользователя на новый, рекомендуемый ему товар. Но для того, чтобы система знала, какой именно товар ей следует рекомендовать, необходимо представить ей данные. Есть следующие способы предоставления данных:

- *Предоставление явных оценок*

Такие оценки предоставляются непосредственно пользователем (например, рейтинг, лайки, обзоры и т.д.).

- *Предоставление неявных оценок*

Такие оценки формируются из взаимодействия пользователя с товаром. Тут могут учитываться кол-во кликов на странице, просмотры пользователя и т.д.

- *Фильтрация по товару*

Используя фильтрацию по товару, система анализирует предпочтения пользователя, исходя из истории его покупок/просмотров и отбирает схожие товары.

- *Фильтрация по пользователю*

Используя фильтрацию по пользователю, система ищет пользователей со схожими предпочтениями/увлечениями, и рекомендует то, что нравится именно им.

После того, как была собрана необходимая информация, система должна измерить схожесть. Схожесть измеряется, используя расстояние из различных метрических систем. Есть множество способов измерения схожести:

1. По метрике Минковского
2. По косинусу угла между двумя векторами, образованных точками данных
3. По коэффициенту корреляции Пирсона
4. По коэффициенту Жаккара
5. По метрике Хэмминга

В современной литературе выделяется три класса рекомендательных систем[4]:

1. Системы, основанные на фильтрации контента. Они предлагают пользователю контент, похожий на тот, который они изучали ранее.

- Например, для рекомендации фильмов можно использовать близость жанров или актёрский состав.
2. Системы, использующие коллаборативную фильтрацию. В этом случае пользователю предлагается контент, заинтересовавший похожих на него пользователей.
 3. Гибридные системы, использующие предыдущие два подхода одновременно.

Практическая часть

Для работы рекомендательного алгоритма необходим набор данных, на который он бы опирался при составлении рекомендаций. У нас было два варианта сбора данных. Первый - создать опрос в Google forms и отправить его друзьям и знакомым. Второй - написать сайт-опрос для сбора данных. Несмотря на простоту первого варианта, мы выбрали второй, так как в этом случае книга однозначно идентифицирована. То есть если участник опроса введёт “Толстой Война и мир”, “Война и Мир Лев Толстой”, алгоритм нашего сайта поймет, что речь идет об одной и той же книге, а вот в Google Forms это было бы два разных варианта ответа.

Сайт был написан на фреймворке Django, так как он позволяет сделать это быстро и просто. После ввода пользователем название книги, алгоритм отправлял поисковой запрос на сайт <https://www.labyrinth.ru>, получал ответ и брал первую книгу, предлагаемую при запросе. После этого сайт сохранял введенную книгу и оценку пользователя в базе данных.

Мы выгрузили сайт в сеть Интернет с помощью бесплатного хостинга “heroku”. Любой человек мог ввести прочитанные им книги и дать им оценку от 0 до 5.

После разработки и загрузки сайта в Интернет, мы попросили знакомых и одногруппников пройти наш опрос (сайт доступен и сейчас по адресу: <https://fathomless-island-49784.herokuapp.com>). Когда его прошли около 30 человек мы начали разрабатывать рекомендательный алгоритм. Предварительно изучив ряд статей[1-3], мы решили что вначале мы напишем алгоритм на основе фильтрации по пользователям (коллаборативную рекомендательную систему).

Алгоритм находит наиболее близкого пользователя к данному пользователю и рекомендует книги, которые ему понравились. Расстояние между пользователями алгоритм находит по евклидово метрике, условно процесс можно разбить на три шага.

1. Найти общие книги у двух данных пользователей

2. Посчитать разницу оценок пользователей по оцененным книгам
3. Вычесть корень из суммы квадратов разностей данных оценок

Рассмотрим это на примере двух пользователей, оценки которых приведены в таблице.

Таблица 2 – Оценки пользователей

	Обломов	Ромео и Джульетта	Преступление и наказание	Отцы и дети
Паша	1	3	5	4
Света	4	-	4	5

1. Общими книгами для Паши и Светы будут “Обломов”, “Преступление и наказание” и “Отцы и дети”.
2. Разница оценок по этим книгам соответственно: 3; 1; 1
3. Расстояние составит $\sqrt{3^2 + 1^2 + 1^2} \approx 3.317$

Также была попытка сделать алгоритм на основе фильтрации по товару со схожим методом работы, однако он оказался значительно медленнее и не удовлетворял пользователей.

Выводы проекта:

1. Были изучены принципы работы рекомендательных систем.
2. Были получены навыки разработки сайтов, используя фреймворк Django, умение вёрстки сайтов с использованием HTML/CSS, а также фреймворка Bootstrap.
3. Были получены навыки программирования на языке Python 3.
4. Была разработана собственная рекомендательная система, удовлетворившая первых пользователей. В перспективе возможна дальнейшая доработка системы и расширение функционала, расширение круга пользователей.

СПИСОК ЛИТЕРАТУРЫ

1. Badreesh Shetty “An in-depth guide to how recommender systems work”, 2019.
2. Baptiste Rocca, Joseph Rocca, “Introduction to recommender systems”, 2019.

3. Pukit Sharma, “Comprehensive Guide to build a Recommendation Engine from scratch (in Python)”, 2018.
4. Волжина Елена Григорьевна, Рекомендательная система для образовательного контента, бакалаврская работа, 2016.
5. Kim Falk, “Practical Recommender Systems”, Manning Publications, 2019.

РАЗРАБОТКА ПРОГРАММЫ ШИФРОВАНИЯ ТЕКСТА НА ЯЗЫКЕ PASCAL

Фролкина Анжелика

МОУ «Средняя школа № 4 городского округа Стрежевой с углубленным изучением отдельных предметов», 10 класс

Томская область, г. Стрежевой

Руководитель: Комарова Ольга Владимировна, учитель математики высшей категории

В современном мире следят за каждым действием человека, нарушается его личное пространство. Тотальная слежка - проблема современного мира, набирающая обороты. Проблема состоит в том, что говорить о приватности и неприкосновенности частной жизни становится все сложнее и сложнее. В качестве противовеса тотальной слежке инженеры разрабатывают защищенные смартфоны, сети, передачу данных, шифрование данных, одежду и маскировку, защищающие от распознавания лиц и образов.

Актуальность моей работы состоит в том, что людям необходимо не дорогое и удобное средство защиты их личной информации

Каждый день тысячи людей, заходя в свой компьютер, обнаруживают, что их личная информация украдена. Хакеры ежеминутно взламывают социальные аккаунты, электронные почты для того, чтобы рассылать спам, распространять вирус по сети, украсть деньги с электронных банковских счетов и т.д. Правил для борьбы с таким мошенничеством не существует, но можно усложнить процесс для злоумышленников, применив криптографические программы как методы защиты информации.

Цель проектной работы:

Создание программы шифрования для использования её в личных переписках.

Задачи проектной работы:

- изучить литературу и Интернет ресурсы для выявления актуальности проблемы;

- познакомиться с наукой «Криптография»;
- исследовать основные методы шифрования;
- написать программу шифрования по одному из изученных методов;
- познакомить одноклассников со своей программой;
- показать необходимость знаний информатики при создании своей программы;
- показать необходимость знаний математики при расчёте бюджета.

Новизна моего проекта в том, что в свободном доступе нет простых и удобных для пользователя программ шифрования данных. Я же предлагаю такую программу для личного пользования.

Прикладной характер проектного продукта:

Программа предназначена для практического использования в повседневной жизни

Выбрав тему проекта и увидев проблему, я стала искать литературу и статьи о науке «Криптография».

Наукой, изучающей математические методы защиты информации путем ее преобразования, является криптология. Криптология разделяется на два направления – криптографию и криптоанализ. Криптография изучает методы преобразования информации, обеспечивающие ее конфиденциальность и аутентичность.

Прочитав книги Брюса Шнайера "Прикладная криптография", "Секреты и ложь. Безопасность данных в цифровом мире", Филиппа Циммермана "Введение в криптографию", я выяснила, что в современном мире люди все чаще обмениваются информацией между собой в интернете, на конференциях или занятиях.

В интернете я нашла бесплатную программу шифрования текстов "Crypto Notepad 1.6.0.0", в ней можно зашифровать заметки, в которых ты записываешь личные данные.

Я прочитала курсовую работу "Разработка программы шифрования", в которой была создана программа для шифрования текста тремя разными видами шифров².

Я сделала вывод о том, что существует много способов зашифровать информацию, и они применяются для разных программ шифрования. Но такие программы применяются только в военных или государственных целях, их мало в свободном доступе для обычных пользователей. Поэтому я решила разработать программу шифрования для личного пользования. Целевая аудитория моего проекта: пользователи ПК (с 12 лет).

Таблица 1 – Ресурсы проекта

Нормативно-правовая база	федеральный закон РФ о защите персональных данных №152 Постановление Правительства Российской Федерации от 16 апреля 2012 г. №313 Приложение к приказу ФСБ РФ от 9 февраля 2005 г. № 66 "Положение о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)"
Финансовые ресурсы	деньги родителей
Информационные ресурсы	интернет, личная библиотека, городская библиотека

Таблица 2 – План работы над проектом

Этап	Содержание	Сроки	Ответственные
Подготовительный	Определить проблему	Сентябрь 2020	Учитель Ученик
Начальный	Изучить литературу по данной теме	Сентябрь-октябрь 2020	Ученик
	Сформулировать цель	Октябрь 2020	Учитель Ученик
	Сформулировать задачи работы	Октябрь 2020	Ученик
Основной	Рассчитать бюджет проекта	Январь 2021	Ученик
	Написать программу шифрования	Ноябрь-декабрь 2020	Учитель Ученик
	Сделать выводы	Январь-февраль 2021	Учитель Ученик
Заключительный	Оформить проектную работу	Февраль 2021	Учитель Ученик

Таблица 3 - Бюджет проекта

№ п/п	Наименование статьи	Источник поступления	Единица измерения	Кол-во	План, руб.	Факт, руб.
1.	Доходы					
1.1	Карманные деньги	Деньги родителей	рубль	1000	1000	1000
2.	Расходы					
2.1.	Оплата электричества	Карманные деньги	час	120	46,12	46,12
3.	Экономия					

3.1	Работа компьютерного мастера и IT-специалиста	Карманные деньги	час	120	72000	0
3.2	Покупка ноутбука	Карманные деньги	штуки	1	10000	0
	Итого:					953,88

При расчете бюджета я учитывала время, затраченное на написание программы, и количество электричества, использованного для работы ноутбука. Также я рассчитала примерную стоимость работы компьютерного мастера и IT-специалиста. Данный расчёт нам показывает, что написание такой программы самостоятельно (без помощи специалистов) дешевле.

Есть программы, использующие шифр Цезаря, но они написаны на других языках программирования, например, Python, C++. Я же использовала шифр Цезаря и язык PascalABC.net и разработала код программы, позволяющий зашифровать сообщение.

Расширение файла программы: .exe, поэтому она должна открываться на любом компьютере. Она последовательно задает пользователю вопросы и шифрует текст на русском языке в зависимости от получаемых данных. Пользователь должен добавить в конце текста цифру сдвига для того, чтобы была возможность расшифровать данный текст. На мой взгляд, это не очень удобно, потому что нужно вручную вводить число. Также пока проблемой является то, что программа не шифрует цифры. В дальнейшем я планирую усовершенствовать программу и упростить ее для пользователя.

```

текст читается с экрана или из файла
1 - с экрана
2 - из файла
1
введите текст
Лучше один раз увидеть, чем сто раз услышать
введите сдвиг
3
1 - зашифровать
2 - расшифровать
1
Зашифрованное сообщение:
Оцъьи сзлр угк целзихя, ъип фхс угк цфоуыгхя
записать сообщение в файл?
1 - да
2 - нет
2
1 - продолжить работу с программой
2 - завершить работу с программой

```

Рисунок 1 – Пример вывода данных на экране

Я считаю, что достигла цели работы – создала программу шифрования. Я выполнила все поставленные передо мной задачи. В дальнейшем я планирую совершенствовать свою программу.

Прикладной характер результата проектной работы

Я опробовала программу на личных сообщениях с одноклассницей. Достигнутый результат доказал удобство и практичность моей программы.

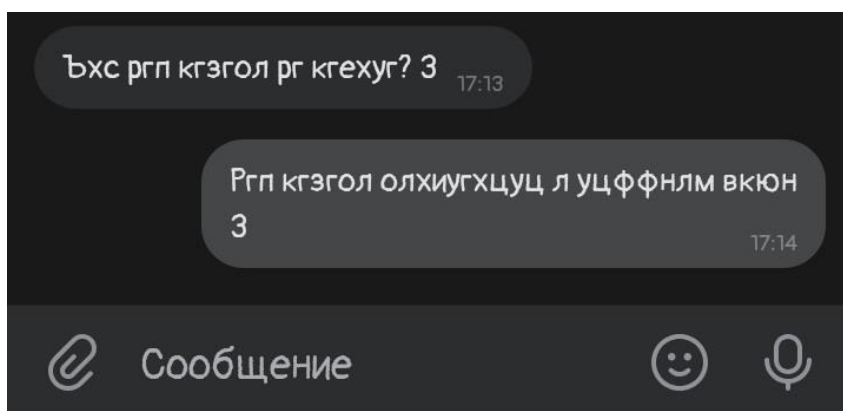


Рисунок 2 – Зашифрованная личная переписка в Вконтакте

Я рекомендую использовать эту программу в личных целях для защиты информации

Выводы

Проблема защиты информации сегодня очень актуальна и останется таковой ещё долго - до тех пор, пока человечество не создаст революционно новые программы защиты всей информации. До тех пор нам не стоит забывать о важности защиты денег и личной информации и пользоваться программами, которые позволят хотя бы чуть-чуть защитить вашу информацию.

В ходе выполнения работы получен готовый продукт - программа шифрования. Полезный эффект при применении программы – защита личных данных от использования другими лицами.

Использование программы может иметь негативные последствия, если человек, пользующийся программой, будет зашифровывать информацию, которая может нанести вред другому человеку.

Рассчитанный бюджет показывает, что самостоятельное написание программы дешевле, чем заказ у специалиста.

В ходе проектной работы я расширила свои знания о языке PascalABC.net, смогла применить свои знания математики, информатики и криптографии на

практике, научилась анализировать ресурсы и риски проекта и планировать дальнейшие действия.

**РАЗРАБОТКА ОБРАЗОВАТЕЛЬНОГО ПРИЛОЖЕНИЯ
«СИМВОЛЫ КУЗБАССА» ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ НА
ОСНОВЕ ОС ANDROID и IOS В ОБЛАЧНОЙ СРЕДЕ
ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ THUNKABLE**

Чибисов Дмитрий

Муниципальное бюджетное образовательное учреждение

«Основная общеобразовательная школа №56», 8 класс

г. Кемерово

Руководитель: Гудченко Вера Тимофеевна, учитель информатики

Актуальность. В современном обществе с появлением разнообразия мобильных устройств (гаджетов) работа с ними и создание различных инструментов и приложений для них становится актуальным не только для специалистов. Мобильные устройства работают на различных операционных системах, но самыми распространенными и открытыми для программирования являются устройства на популярной платформе Android.

Программирование мобильных приложений на платформе Android может пригодиться для разработки очень широкого круга программных систем, начиная от игровых приложений для мобильных телефонов, и заканчивая профессиональными системами, использующими современные технологии.

Кроме того, современное общество, использующее информационно-коммуникационные технологии и разнообразные устройства: от мобильных устройств до бытовой и домашней техники, требует наличия навыков программирования.

Новизна. В обычной школе уделяется очень мало времени обучению основам программирования, развитию интереса учащихся к практической работе с мобильными устройствами и программами, формированию умений и навыков для осуществления творческой деятельности в области создания мобильных приложений под ОС Android и использование технических средств ИКТ в повседневной жизни.

Проблема. Недостаточное знание символов Кузбасса детьми и взрослыми – жителями Кузбасса.

Цель проекта – использование облачной среды визуального программирования Thinkable для создания образовательного мобильного приложения «Символы Кузбасса» на платформе Android.

Для достижения поставленной цели нужно решить следующие **задачи**:

- познакомиться с основами алгоритмизации;
- узнать облачную среду визуального программирования Thinkable и принципы создания в ней мобильных приложений;
- понять геральдику и её законы;
- познакомиться с гербами городов Кузбасса;
- научиться проектировать мобильные приложения, создавать программы и выполнять их отладку на мобильных устройствах.

Теоретическая значимость нашей работы заключается в анализе теоретического материала о визуальной среде программирования Thinkable.

Мы привыкли использовать смартфон для связи и игр, но вообще он способен на большее. Например, научить нас чему-нибудь полезному.

Практическая значимость заключается в том, что создание нами мобильного приложения будет способствовать:

1. знакомству и изучению символов Кузбасса детьми и взрослыми;
2. популяризации облачной визуальной среды программирования Thinkable.

Thinkable (<http://thinkable.com>) - это облачная среда программирования, разработанная учеными Массачусетского Технологического Института (MIT), предназначена для разработки приложений для мобильных устройств (смартфонов и планшетных компьютеров), работающих на операционной системе Android и IOS.

Вместо текстового языка программирования, в таких средах используются визуальные логические блоки с уже готовыми действиями, функциями для работы с социальными сетями, веб-сайтами или сенсорами устройства и др. Визуальная среда разработки позволяет обрабатывать мультимедийный контент, распознавать речевые команды и синтезировать речь.

Что нам понадобится:

1. Подключенный к интернету компьютер с установленным на нем браузером (например, GoogleChrome, MozillaFirefox или Safari). С помощью браузера мы будем создавать наше приложение. Дополнительно на компьютер ничего устанавливать не нужно.

2. Смартфон или планшетный компьютер с операционной системой Android 4.0 или выше.
3. Аккаунт Google или GoogleApps - он нам будет нужен для того, чтобы авторизоваться на сайте MIT AppInventor.

Вновь создаваемое приложение в среде Thinkable - это новый проект. После входа в Thinkable пользователь попадает на страницу, где отображается список его проектов. Все созданные проекты хранятся в папке Мои проекты.

Разработка мобильного приложения в Thinkable происходит в 2 этапа. *Первый этап* - проектирование интерфейса пользователя «Как это будет выглядеть», *второй* - программирование компонент приложения «Как они будут себя вести».

Интерфейс языка программирования Thinkable состоит из двух основных частей - дизайнера и редактора блоков.

В дизайнере мы конструируем наше приложение из элементов - экранов, кнопок, ячеек, изображений, звуков. В редакторе **блоков** мы программируем поведение этих элементов.

Разработка приложения происходит в облачной среде Thinkable. **Тестирование и отладка происходят на мобильном устройстве.**

Среда англоязычная (на данный момент). Одно из главных преимуществ данной среды перед аналогичными сервисами - очень удобный способ тестирования готовых проектов: нет необходимости загружать файл проекта на мобильное устройство и устанавливать все промежуточные версии проекта, чтобы проверить его работу. Достаточно лишь воспользоваться мобильным приложением Thinkable, которое можно установить через Google Play или App store. Мобильное приложение синхронизируется с web-версией сервиса и позволяет тестировать проекты сразу в процессе создания.

Материально-техническое обеспечение проекта:

- Планшеты или мобильные устройства с ОС Android.
- Выход в Интернет.

Схема создания проекта:

№	Этап	Деятельность
1.	Генерация идей	Разработка различных идей создания приложения, анализ предложенных идей с точки зрения возможных потребителей приложения и потребности его создания.

2.	Разработка прототипа	Разрабатывается прототип приложения, который содержит необходимые компоненты и демонстрируется, как в целом будут выглядеть все экраны приложения.
3.	Программирование	Создаётся код приложения с помощью Blockly.
4.	Внедрение	Проверяется созданное приложение на возможность анализа другими пользователями, присваивается всем переменным, процедурам и компонентам программы идентификаторы, переименовываются компоненты, если необходимо добавляются комментарии программного кода.
5.	Тестирование приложения и фиксация ошибок	Установка приложения на мобильные устройства, тестирование его и отладка.
6.	Оформление	Создание описания приложения, установка названий экранов.
7.	Оценка	Создание краткого описания. <i>Событие</i> представляет собой некоторое действие, которое активизирует стандартную реакцию компонента. В качестве события могут рассматриваться нажатие кнопки, выбор пункта меню. <i>Действия</i> - операции или функции, которые деталь компонент может выполнять. Действия могут быть инициированы другими деталями посредством соответствующих связей.
8.	Дизайн	Разработка дизайна приложения включает разработку эскизов интерфейса пользователя для каждого экрана. На эскизе каждого экрана, желательно аннотировать, что делает каждый размещенный на нем компонент. Разработка иконки приложения. Установка иконки для приложения.

Описание приложения «Символы Кузбасса». Приложение «Символы Кузбасса» позволяет пользователю познакомиться с гербами городов Кузбасса, услышать их описание и проверить себя по этой теме.



Рисунок 1 – Приложение «Символы Кузбасса»

В результате выполнения проекта необходимо создать дизайн приложения в режиме «Дизайн» и запрограммировать действия для каждого компонента, в режиме «Блоки». Используемые компоненты:

1. Надпись - компонент, который используется для вывода фрагментов текста на экран.
2. Кнопка - компонент, при нажатии на который будет появляться текстовый файл.
3. Изображение - компонент, позволяющий отображать на экране графические объекты.

Приложение включает в себя двадцать три экрана: Screen1-Кузбасс, Screen2-Карта, ..., Screen 23-Проверка, на которых размещены компоненты: надписи, кнопки и изображения.

Заключение. Среда программирования Thinkable *позволяет любому создавать красивые и мощные образовательные мобильные приложения.* и нацелена на развитие интереса учащихся к практической работе с мобильными устройствами и программами, формирование представлений об основных правилах и методах программирования мобильных устройств, развитие у учащихся логического мышления, конструкторских способностей в процессе моделирования и экспериментов.

Творческая проектная деятельность по созданию приложений для мобильных устройств позволяет наглядно увидеть результаты своей работы и оценить полезность и значимость развития навыков программирования для жизни.

«SMARTHAND из LEGO MINDSTORMS EV3»

Якимишина Анна Петровна

МАОУ средняя общеобразовательная школа №2 с. Александровское

10 класс

Александровского района Томской области

Руководитель: Баротов Н. М, учитель физики и робототехники

Аннотация

Мир роботов имеет огромное значение в жизни человечества. Каждый день появляются все новые и новые технологии которые помогают, облегчают нам обыденную жизнь. Изучении робототехники вызвало во мне интерес уже со времен начальной школы. В этой работе я собрал результаты моих долгих учений. Я собрал робота «умную руку» из набора Lego Mindstorms ev3. Запрограммировала робота на языке программирования LEGO MINDSTORMS Education EV3. Получившийся конструкция одевается на руки и становится как бы продолжением руки, но с уникальными возможностями. Именно поэтому проект назван **SMARTHAND**, что в переводе означает умная рука.

Введение

Наука не стоит на месте. Уже сейчас уровень развития робототехники достиг больших высот. Писатели-фантасты неоднократно пугали мир разнообразными вариациями на тему «бунта машин». Но ситуация с развитием робототехники в настоящее время складывается таким образом, что остановить это развитие в данной сфере невозможно. А все потому, что роботы уже заняли свою нишу в жизни общества. Они стали частью современной промышленной революции, характеризующейся широким внедрением адаптивных технологий и роботизацией производства. Ежегодно все больше предприятий автоматизируется, поэтому на данный момент завод, на котором работает всего несколько десятков человек, а всю основную работу выполняют роботы, уже никого не удивляет. Промышленные роботы выпускаются десятками тысяч. Несмотря на то, что этот рынок достаточно давно сформировался, с выходом на него Китая ситуация здесь только обостряется.

Сегодня нам сложно представить свою жизнь без роботизированных технологий. Нас везде окружают роботизированные механизмы. Конечно, это хорошо, ведь без роботов наша жизнь была бы, намного, сложно, чем можно представить себе. Я изучаю мир роботов уже несколько лет. В нашей школе есть почти все необходимое для полноценного развития в технологическом плане.

Актуализация

Актуальность данной работы связана с предметной частью исследования. Робототехника является важнейшим аспектом в современной жизни человечества. Мы представить не можем свою жизнь без современных роботизированных технологий. Развивая робототехнологию, мы развиваем свою жизнь, свою реальность. Машины были и будут актуальными в нашей жизни. Мой проект можно использовать как в образовательных, так и в исследовательских целях.

Цель: разработать с помощью Lego mindstorms ev3 наиболее удобную, свободно манипулируемую, автоматизированную техническую систему дистанционного управления роботом.

Задачи:

- Изучить основы робототехники, набор Lego Mindstorms
- Изучить принципы гироскопического датчика и моторов Lego
- Научиться собирать конструкции
- Научиться написать программу для робота на языке LEGO MINDSTORMS Education EV3.
- Разобраться с настройками в блоке управления робота

Гипотеза: Собрать конструкцию из LEGO MINDSTORMS Education EV3 для дистанционного управления робота, частично управляемое жестами.

Основная часть

Значимости развития робототехники в жизни человечества.

Главное предназначение робототехники - сделать жизнь человека наиболее удобной, увеличить производительность рабочих процессов, за счет улучшений условий труда. Использование роботов распространено практически во всех сферах человеческой жизни. Не редко удается наблюдать роботов в промышленности, где они автоматизируют большую часть производственных задач. С каждым днем развития новых технологий приобретают все больше развития. То что вчера нам казалось сказкой на сегодняшний день является нашей реальностью. Еще пару лет назад

искусственный интеллект казался чем то невообразимым, а сегодня его использование широко распространено в жизни человечества. Роботизация продолжает преобразовывать основные аспекты нашей жизни в том числе, здравоохранение и образование. Роботы проектируются чтобы, быть частью экономики мира — упростить повседневную жизнедеятельность людей, помогая обществу акцентировать внимание на улучшение товаров и услуг.

Идея создания проекта

Увидев, как ребенок, играя с машинкой, управляет ей дистанционно, с помощью пульта, мне захотелось попробовать воссоздать что-то подобное с помощью набора LEGO MINDSTORM EDUCATION EV3. В процессе работы я задумалась: что еще может заставить робота совершать определенные действия, что может сделать управление проще и интереснее. В этот момент мне и пришла идея по созданию Smarthand.

Процесс сборки, программа и алгоритм работы Smarthand-a.

Внешний вид:



Рисунок 1 – Процесс сборки робота

Процесс сборки заняла около 2 недель времени. Я пробовала несколько вариантов конструкций проекта. Остановилась на этом варианте так как мне было комфортно двигать пальца для управления роботом.

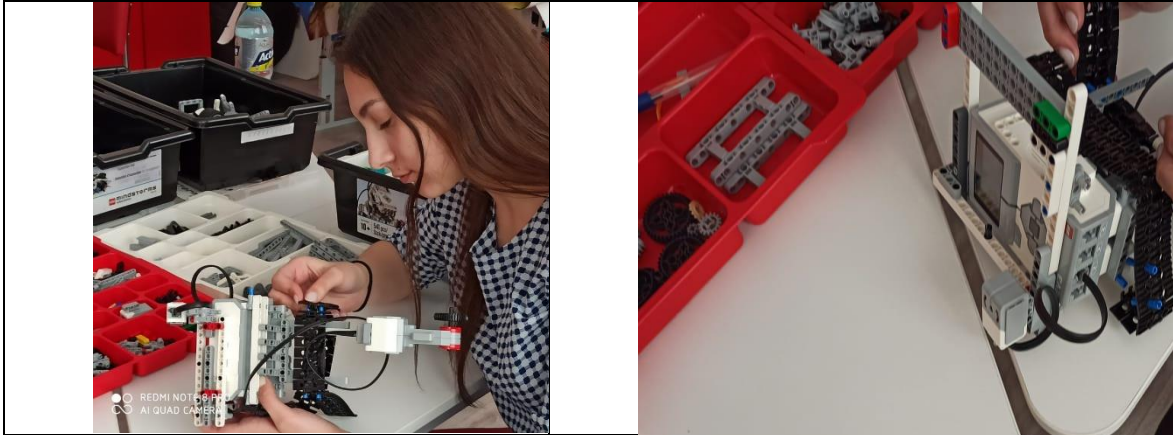


Рисунок 2 – Процесс сборки робота

Для использования данного устройства необходимо:

1. Подключить между собой блоки управления Smarthead и робота через Bluetooth
2. Запустить программу на роботе.
3. Запустить программу на Smarthead

Программа для Smarthead выглядит следующим образом:

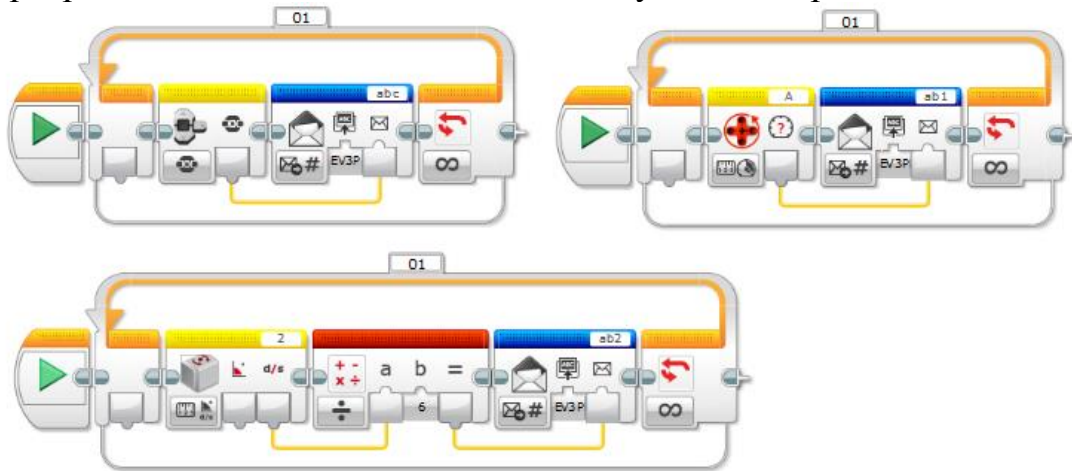


Рисунок 3 – Программа для Smarthead

Программа для робота (которым будут управлять Smarthead-ом) :

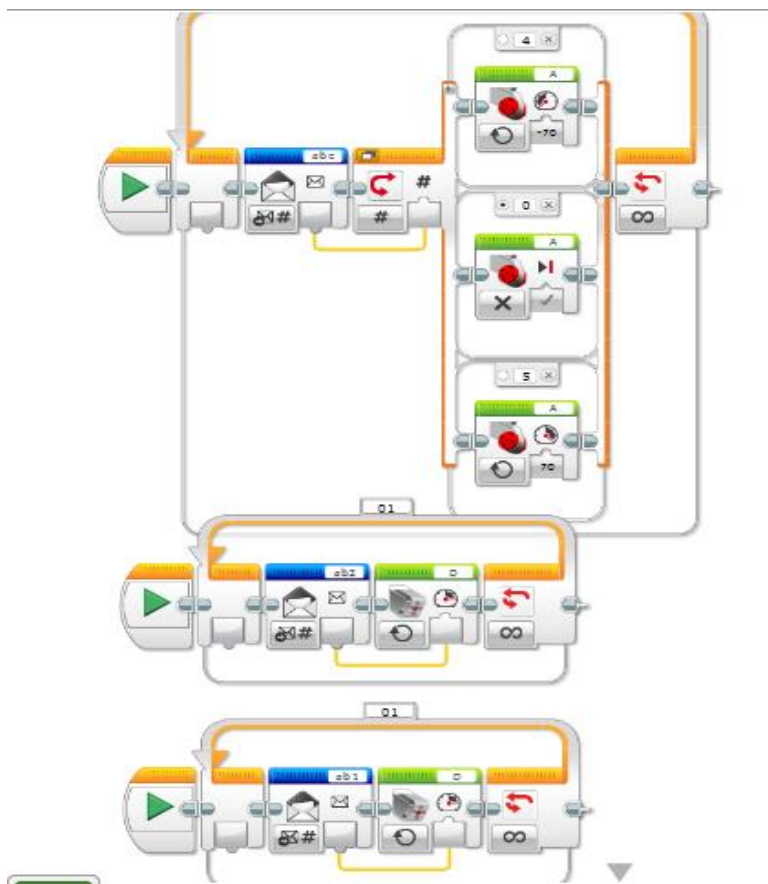


Рисунок 4 – Программа для робота

Робот обязательно должен содержать в себе следующие детали:

1. Большой мотор - отвечающий за движение вперед, назад.
2. Средний мотор – отвечающий за рулевое управления.

Алгоритм работы

Блок управление Smarthand, через блютуз передает данные на блок управление робота. А именно датчик гироскопа передает данные об изменении угла наклона на что реагирует средний мотор робота. А за движении вперед назад отвечает кнопки вверх и вниз блока управления.

Вывод

Делая вывод о проделанной работы могу сказать, что поставленные цели мною были достигнуты. Работая над этим проектом я углубил свои знания в области программирования. Мы разработали с помощью набора Lego mindstorms ev3 наиболее удобную, свободно манипулируемую, автоматизированную техническую систему для дистанционного управления роботом.

Все поставленные мною задачи были достигнуты. А именно мне удалось :

- Изучить основы робототехники, набор Lego Mindstorms EV3
- Изучить принципы гироскопического датчика и моторов Lego
- Научиться собирать конструкции
- Научиться написать программу для робота на языке LEGO MINDSTORMS Education EV3.
- Разобраться с настройками в блоке управления робота

СПИСОК ЛИТЕРАТУРЫ

1. Физика. 7кл. : учебник/ А. В. Перышкин. – 4-е изд., стереотип. – М. : Дрофа, 2015. – 224с. : ил. (дата обращение 05.01.2021)
2. Электронный ресурс, ссылка доступа: (<https://yandex.ru/images/search?text=как%20вычислить%20площадь%20обуви&stype=image&lr=67&source=wiz>), (дата обращение 15.01.2021)
3. Официальный сайт RESHAK.ru, ссылка доступа (<https://reshak.ru/otvet/reshebniiki.php?otvet=zad/36&predmet=perishkin7>) (дата обращение 18.01.2021)

ЯЗЫК ПРОГРАММИРОВАНИЯ SLAVASCRIPT

Шестаков Вячеслав Григорьевич

МАОУ КУГ №1 – Универс, 9 класс

Руководитель: Вахитова Екатерина Юрьевна, учитель информатики

Актуальность

Синтаксисы существующих языков программирования несовершенны. В результате данного проекта будет разработан язык, в котором некоторые недостатки языков высокого уровня будут устранены. В данном проекте был создан язык программирования, объединяющий Python и C++, но при этом с более удобным синтаксисом, чем в исходных языках.

Цели и задачи

Цель: создание языка программирования, объединяющего синтаксис Python и C++ и имеющего более удобный синтаксис.

В данном проекте требовалось выполнить много задач, часть из которых заключалась в изучении материалов, часть в написании своего языка программирования, а другая в написании документации.

Задачи:

- Изучить теорию по написанию языков программирования
- Проанализировать существующие языки программирования высокого уровня и их синтаксис.
- Придумать синтаксис будущего языка и его названия.
- Создать язык программирования и протестировать его.
- Убрать проблемы, возникшие в процессе создания языка.
- Перенести язык программирования на телефон.
- Написать документацию.
- Добавить документацию в мобильную версию языка.

Теоретическая часть

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы.

Лексический анализатор получает на вход исходный код языка программирования и выдаёт список токенов, то есть единицы языка: переменные, названия функций (идентификаторы), операторы, числа, строки и ключевые слова. После работы лексера не будет совершаться обращение к исходному коду, а будет использоваться только полученный список токенов.

Синтаксический анализатор получает на вход список из токенов, который был сгенерирован лексером, и возвращает абстрактное синтаксическое дерево (в английском – AST – abstract syntax tree), которое позволяет отображать зависимости одних выражений от результата вычисления других выражений.

Анализ существующих языков

Для анализа были выбраны в связи со своей популярностью следующие языки программирования:

Таблица 2 -сводная таблица типов данных

Язык программирования	Недостатки синтаксиса	Преимущества синтаксиса
C++	Строгая типизация	Свободное приведение типов компилятором
Python	Строгая табуляция	Динамическая типизация
Java	Сложный ввод-вывод данных	Проработанный синтаксис для классов
JavaScript	Браузерный язык	Использование Unicode символов

В ходе анализа существующих языков программирования были сформулированы следующие требования к разрабатываемому языку программирования:

- Код всей программы может быть написан вне функции (как в Python).
- Выделение блока будет осуществляться фигурными скобками (как в C++).
- Будет отсутствовать объявление типов переменных (динамическая типизация, как в Python).
- Круглые скобки будут необязательными (как в Python).
- Символы, завершающие выражения, будут отсутствовать (как в Python).
- Строки будут объявляться только в двойных кавычках (как в C++).

Создание основы для языка программирования

Языки программирования не могут существовать без чёткого определения синтаксиса, следовательно требуется определить синтаксис будущего языка программирования.

Рассмотрим программы, которые в зависимости от пользовательского ввода выводят “Big”, “Small” или “Zero” (при числе большем нуля, меньше него и нуле соответственно). Для примера возьмём реализацию этой задачи на Python и C++:

	C++:
	1. <code>#include <iostream></code>
	2. <code>using namespace std;</code>
	3. <code>int main(){</code>
	4. <code>int a;</code>
	5. <code>cin >> a;</code>
	6. <code>if (a > 0) cout << "Big" << endl;</code>
	7. <code>else if (a < 0) cout << "Small" << endl;</code>
	8. <code>else cout << "Zero" << endl;</code>
	9. <code>return 0;</code>
	10. <code>}</code>
Python:	
1. <code>a = int(input())</code>	
2. <code>if a > 0 : print('Big')</code>	
3. <code>elif a < 0 : print('Small')</code>	
4. <code>else : print('Zero')</code>	

Рисунок 1 – Сравнение синтаксиса языков программирования

Из приведённого выше примера видны недостатки синтаксисов этих языков программирования. Например, обязательная постановка двоеточий и точек с запятой, что в свою очередь служит причиной множества ошибок начинающих разработчиков. В связи с этими недостатками код программы на разрабатываемом языке, решающую данную задачу должен выглядеть так:

Будущий язык:

1. `import types`
2. `a = double(input())`
3. `if a > 0 println "Big"`
4. `elif a < 0 println "Small"`
5. `else println "Zero"`

Рисунок 2 – Пример синтаксиса языка SlavaScript

Справка

SlavaScript — интерпретируемый, динамически типизированный язык программирования общего назначения.

SlavaScript поддерживает процедурное и объектно-ориентированное программирование. SlavaScript имеет небольшую стандартную библиотеку, включающую в себя операции ввода-вывода и некоторые алгоритмы.

Существует и постоянно обновляется синтаксис SlavaScript для Notepad++.

Существует версия SlavaScript для телефонов Android, имеющая некоторые особенности в связи с платформой.

Расширение файлов с исходным кодом SlavaScript — «.s++».

Практическая часть

В данном проекте практической частью было создание интерпретатора для SlavaScript. Интерпретатор состоит из лексера и парсера.

Лексер

Лексером является класс `Lexer`, принимающий на вход строку с исходным кодом. Возвращает лексер массив `Token`. В каждом `Token` хранится его тип (`TokenType`), строковое содержимое, номер строки и позиции в ней места встречи данного `Token`. В зависимости от `TokenType` строковое содержимое может присутствовать, или быть пустой строкой.

Разбиение на `Token` происходит путём поиска в исходном коде нужного начала и последующей токенизации данного `Token` (например, при встрече двойной кавычки начинается токенизация `ТЕХТ`). После чего данные `Token` добавляются в список токенов, составляющий весь исходный код программы. При работе лексера все комментарии пропускаются и не добавляются в итоговый список.

Парсер

Парсер в SlavaScript представляется классом `Parser`, принимающим на вход список `Token`, и возвращающим `BlockStatement`, представляющий весь

код программы, как единое блочное выражение. В BlockStatement хранится список из Statement. В каждом Statement могут храниться другие Statement или Expression. Парсинг списка Token происходит методом рекурсивного спуска. Данный рекурсивный спуск происходит путём вызова следующих функций, так, что каждая функция или возвращает значение туда, где она была вызвана, либо вызывает следующую функцию снизу.

Запуск программы

Перед непосредственным запуском программы происходит добавление всех функций, опеределённых в исходном коде и подключение всех импортируемых файлов, что осуществляется по средствам шаблона проектирования Visitor. После чего запуск программы выполняется классом Parser, выполняющим Statement, хранящиеся в BlockStatement программы. При запуске каждого Statement вызывается функция eval, переопределённая для каждого Statement по-своему.

Типы данных

Таблица 2 -сводная таблица типов данных

Тип	Байт	Диапазон значений
Нулевой тип		
Null	1	Хранимое значение отсутствует
Логический тип		
Bool	1	0 / 255
Числовой тип (с плавающей запятой).		
number	∞	∞
Строковый тип		
String	∞	Строки из ASCII символов.
Массивный тип		
Array	∞	Массив с любыми типами данных.
Отображение		
Map	∞	Хранит пары из любых типов данных.
Тип функций		
function	∞	Хранит определение функции.
Тип классов		
Class	∞	Хранит определение класса.

Сравнение кодов на C++, Python и SlavaScript

Для сравнения SlavaScript с C++ и Python решим несколько задач с сайта acmp.ru. Программа 1. Задача №68, 21% сложности.

SlavaScript:

```
import types
s = input()
n = int(input())
if s == "School" && n % 2 == 0 print " "
else print "Yes"
```

```
s = input()
n = int(input())
if s == "School" and n % 2 == 0:
    print("No")
else:
    print("Yes")
```

Python:

Программа 2. Нахождение первых 20 чисел Фибоначчи.

SlavaScript:

```
def fib(n){
    if n <= 1 return n
    else return fib(n - 2) + fib(n - 1)
}
for i = 1, i < 21, ++i println fib(i)
```

C++:

```
#include <iostream>
using namespace std;
int fib(int n){
    if (n <= 1) return n;
    else return fib(n - 2) + fib(n - 1);
}
int main(){
    for(int i = 1, i < 21, ++i){
        cout << fib(i) << endl;
    }
    return 0;
}
```

Вывод

В данном проекте мне удалось придумать синтаксис для нового языка программирования и реализовать интерпретатор для него. Также удалось создать приложение на Android, в котором реализован язык программирования и документацию к нему. В ходе данной работы был получен бесценный опыт, который пригодится для реализации задуманных языковых единиц.

НЕЙРО-СТИЛИЗАЦИЯ ИЗОБРАЖЕНИЙ

Яковлев Павел

Нетиповое муниципальное бюджетное общеобразовательное учреждение

Анжеро-Судженского городского округа «Гимназия №11»,

10 класс

г. Анжеро-Судженск

Руководитель: Мангазеев Александр Юрьевич, учитель высшей категории, программист, тиктокер, пасечник, стример, магистрант ТГУ

Актуальность

Искусственный интеллект давно вышел за рамки технологий далекого будущего и находит широкое применение в настоящее время во всех сферах

жизни общества. Он стремительно развивается, технологии, применяемые 10 лет назад, не сравнимы с современными. На сегодняшний день развлечения и социальные сети являются неотъемлемой частью нашего досуга. Мною было принято решение затронуть обе эти сферы: развлечений и нейросетей - и объединить их в своём проекте.

Цель: Создать Telegram бота для преобразования изображений с использованием генеративно-состязательных нейронных сетей

Задачи:

- Изучить и реализовать архитектуру CycleGAN
- Выбрать задачи преобразований изображений
- Найти и собрать данные под требуемые задачи
- Адаптировать архитектуру для исполнения ботом
- Разместить бота на сервере

Обзор аналогов

В данный момент аналогом данного проекта является Telegram бот Prisma для обработки фотографий, но поддержка данного бота, видимо, прекратилась либо в данный момент приостановлена. Существует также несколько приложений с подобным функционалом, но многие пользователи не желают устанавливать сторонние приложения для возможности обработки и преобразования фотографий и испытывают удовольствие от пользования ботом по преобразованию фотографий без сторонних программ.

Целевая аудитория

Данный проект является полностью коммерческим и доступен абсолютно для всех пользователей мессенджера Telegram. Бот будет интересен всем, желающим просто развлечься или же вскоре профессионально обрабатывать изображения и добавлять им стили. Целевая аудитория данного проекта обширна – от инстаграм блогеров до людей, желающих увидеть свою прекрасную фотографию в новом стиле.

Среды разработки, методы

Для осуществления проекта понадобился бесплатный облачный сервис *Google Colaboratory*, на котором можно обучить искусственный интеллект и не прибегать к покупке мощных видеокарт. Для поиска данных был использован сервис Kaggle. Для загрузки бота на виртуальную машину понадобился GitHub. Для обучения моделей необходимо использовать метод машинного обучения глубокое обучение (Deep learning) с учителем (Supervised learning).

Этапы выполнения проекта

Сбор данных и подготовка

Все данные взяты из открытых источников: kaggle, google. Для выполнения задач понадобилось несколько датасетов: селфи-датасет[5] (изображения с лицами людей), датасет с изображениями мультяшного героя, датасет с карандашным стилем портретного изображения, датасет с картами. Не все данные были готовы к обучению, поэтому необходимо вручную отобрать и подготовить изображения перед обучением.

Построение архитектуры CycleGAN

Для построения архитектуры используем библиотеку PyTorch на языке Python. Для большинства архитектур преобразования изображения в изображение требуется большой набор попарных примеров, подготовить такой набор очень сложно и в некоторых задачах практически невозможно.

Поэтому в нашем проекте используется CycleGAN (циклическая генеративная состязательная сеть)

CycleGAN является дополнение архитектуры GAN, в ней задействовано сразу 2 генератора и 2 дискриминатора.

Каждый генератор используется для непосредственного преобразования изображений, а дискриминаторы – для оценки их правдоподобности[4].

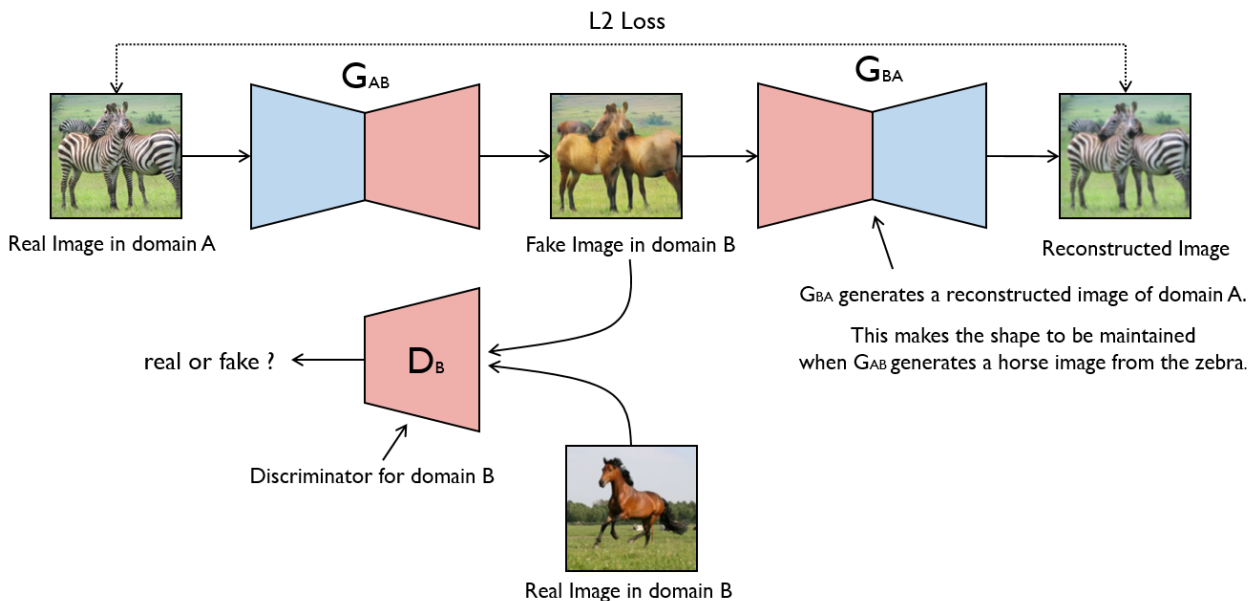
(Весь код в репозитории Github

<https://github.com/PoulYak/StrangeFeaturesBot>)

Обучение генеративно-состязательной сети

Глубокие нейронные сети состоят из нескольких слоёв, каждый из них во время обучения способен захватывать определенную информацию с изображения в зависимости от того, на какой глубине находится слой и передавать оставшиеся данные следующему. Информация, извлекаемая глубокими слоями имеет решающее значение, в этих слоях содержится суть изображения вне зависимости от контекста.

Особенность CycleGAN в том что при обучении мы будем обучать 2 генератора для преобразования изображений. Стоит учитывать, что на обучение тратится достаточно много времени



Создание бота Telegram

Для создания бота была использована библиотека `pyTelegramBotApi`. Создание бота именно в Telegram обосновано удобством работы с методами, предоставляемыми разработчиками Telegram и его стремительным ростом в популярности использования. Бот функционирует по технологии `long pool`, в перспективе переделать это использование методов `webhook`[2].

Размещение бота

Для размещения бота было несколько вариантов платформ: Heroku, Google Cloud, AWS

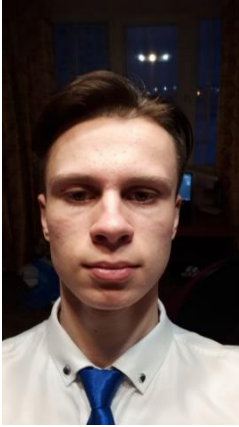
При попытке размещения на некоторые из них возникли проблемы: Heroku сервисы не могут проводить такие вычисления, при попытке создания аккаунта на Google Cloud Platform, видимо, по законам РФ не разрешено создавать аккаунт для личного пользования.

Поэтому без лишних проблем бот был размещён на виртуальной машине AWS с EC2[3]. Для передачи файлов с локального компьютера на VM (virtual machine) был использован крупнейший веб-сервис для хостинга IT-проектов GitHub. Для успешного запуска бота на виртуальной машине понадобилось использовать менеджер терминалов `tmux`[1].

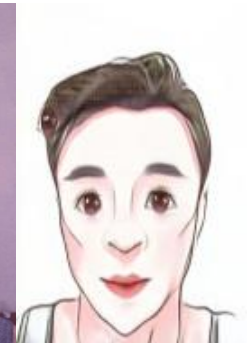
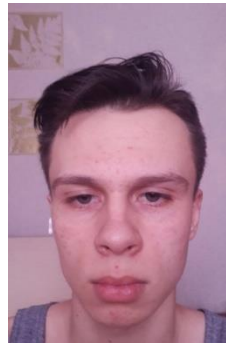
Представление результата преобразований бота

(Ссылка на бота в Telegram <https://t.me/StrangeFeaturesBot>)

Карандашный стиль



Мультяшный стиль



Преобразование изображений со спутника в карты



Заключение

Цель моего проекта достигнута, задачи в ходе работы выполнены полностью. Бот для стилизации изображений запущен и уже успешно работает на протяжении нескольких недель.

Бюджет проекта состоит из пользования Google Colab (бесплатно), размещения virtual machine on AWS EC2 (бесплатно), при увеличении вычислительной мощности будут затраты, также данные были собраны вручную с открытых источников, поэтому на них затраты отсутствуют.

Ссылка на бота в Telegram <https://t.me/StrangeFeaturesBot>

Ссылка на репозиторий GitHub

<https://github.com/PoulYak/StrangeFeaturesBot>

Вывод

Удалось изучить и реализовать архитектуру CycleGAN с помощью библиотеки PyTorch на языке Python. Были выбраны задачи по преобразованию снимков со спутника, применение мультяшного и карандашного стилей. Удалось найти и собрать данные под требуемые задачи.

Адаптация архитектуры для исполнения ботом получилось и в итоге бот был успешно размещён на виртуальной машине с помощью сервиса AWS.

СПИСОК ЛИТЕРАТУРЫ

1. Шпаргалка по TMUX [Электронный ресурс] <https://losst.ru/shpargalka-ro-tmux> (дата обращения 14.02.2021).
2. Пишем ботов для Telegram на языке Python [Электронный ресурс] <https://groosha.gitbook.io/telegram-bot-lessons/> (дата обращения 01.02.2021).
3. How To Host Telegram Bot on AWS - Amazon Web Services [Электронный ресурс] <https://youtu.be/MArTRT37B0k> (дата обращения 05.02.2021).
4. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Электронный ресурс] <https://youtu.be/m47qsfSZoTI> (дата обращения 14.01.2021).
5. Selfie dataset [Электронный ресурс] <https://www.crcv.ucf.edu/data/Selfie/> (дата обращения 20.01.2021).

СОЦИАЛЬНО ОРИЕНТИРОВАННАЯ КОМПЬЮТЕРНАЯ ИГРА «БЕРЕГИТЕ ПРИРОДУ» В ПРОГРАММЕ UNITY

Заровняев Айаал

МАОУ Саха политехнический лицей ГО «город Якутск», 8 класс,

Руководитель: Стручкова А.Н., учитель географии

Актуальность: В настоящее время в связи с массовым загрязнением окружающей среды всё более остро стоит вопрос экологического

воспитания детей. Одним из путей решения вышеуказанной проблемы может стать использование электронных устройств, направленных на привитие грамотного отношения к природе. Следовательно, возникает необходимость создания продукта, отвечающего критериям мобильности, познавательности и экологичности. В связи с этим разработана социально-ориентированная игра «Берегите природу» которая приспособлена к мобильным устройствам (планшеты и телефоны)

Игра будет актуальной, потому что в наш век пластиковой посуды и полиэтиленовых упаковок, игра будет не только развлекать ребенка, но и учить, что нельзя выбрасывать пластиковую упаковку в лесу, то есть беречь природу.

Объект исследования: программирование на платформе «Unity».

Предмет: процесс создания компьютерной игры «Берегите природу».

Гипотеза: уровень экологической грамотности детей можно повысить, если реализовать социально-ориентированную игру «Берегите природу».

Цель: создание социально-ориентированной компьютерной игры «Берегите природу» для повышения экологической грамотности детей.

Задачи исследования:

1. Изучение литературы;
2. Отбор содержания игры;
3. Изучить программу по разработке игр Unity;
4. Разработка программы игры (персонаж, правила, дизайн);
5. Апробация игры.

Методы:

1. Теоретические (сравнение, прогнозирование, структурирование).
2. Эмпирические (программирование)

Этапы работы:

1. Теоретический этап

- 1.1. Изучение экологических проблем
- 1.2. Изучение компьютерной программы Unity.

2. Экспериментальный этап

- 2.1. Отбор содержания (классификация типов мусора).
- 2.2. Процесс создания игры.
- 2.3. Апробация.



Типология глобальных проблем (по В.П. Максаковскому)

Способы переработки отходов

- естественное разложение в природной среде

1. Пищевые отходы — срок разложения 30 дней.
2. Листья, семена, веточки — срок разложения 3-4 месяца.
3. Офисная бумага — срок разложения 2 года.
4. Железные банки, старая обувь, текстиль — срок разложения 10 лет.
5. Обломки кирпича и бетона — срок разложения 100 лет.
6. Электрические батарейки — срок разложения 110 лет.
7. Пластиковые бутылки — срок разложения 180-200 лет.
8. Алюминиевые банки — срок разложения 500 лет.

- захоронение на полигонах

- рециклинг (выделение полезных компонентов и вторичная переработка)

- Механический рециклинг; нарезка и измельчение отходов, с последующим использованием их в качестве наполнителей для новых материалов.
- Инсинерация; сжигание отходов с целью получения энергии.
- Рециклинг пиролизом; сжигание в безвоздушной среде.
- Химический рециклинг; обработка химическими реагентами.

Среднестатистический россиянин **производит** в год около двух кубометров **мусора** — примерно 400 кг. Это около 1,1 кг в день.

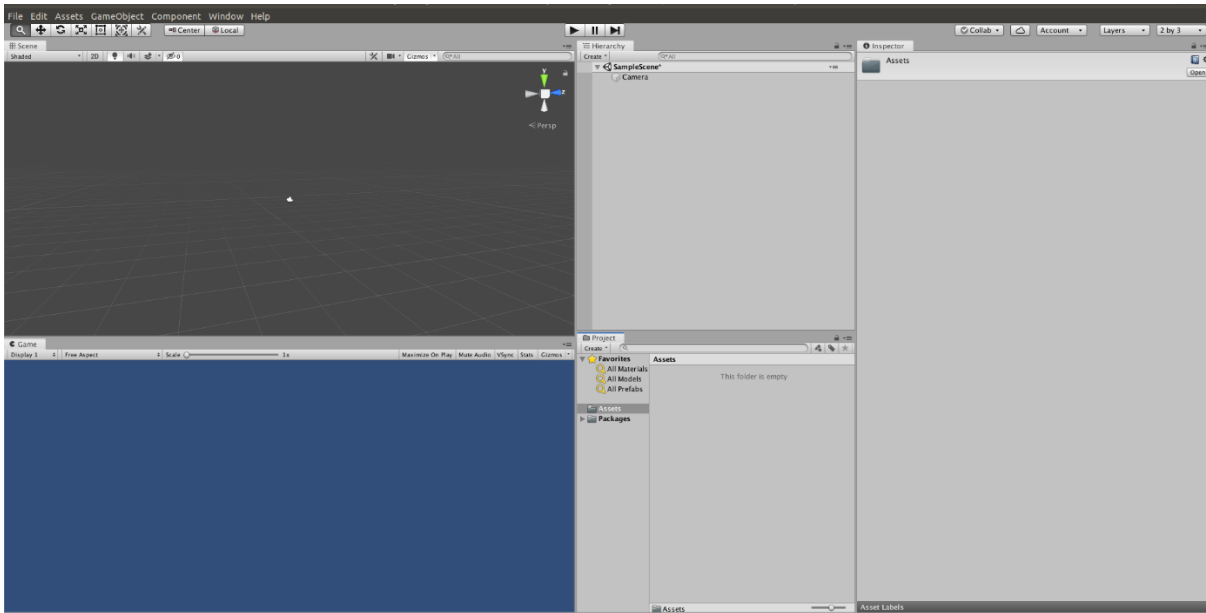
На данный момент в городе Якутске действует только 2 точки для раздельного сбора мусора. В 2021 году за счет федеральных средств будут закуплены и установлены контейнеры в разных районах города, на эти цели будет выделено 40 млн рублей.

Unity— это движок, программа, среда разработки, которая помогает создавать игры. Его создали в 2005 г. в компании Unity Technologies в Сан-Франциско. Первоначально движок работал только под Mac, но сейчас Unity поддерживает больше 25 платформ: Windows, Linux, Android, iOS и др.

Движок поддерживает два скриптовых языка: C#, JavaScript (модификация). В данном проекте будем использовать язык C#, так как он является удобным и понятным в использовании.

Основные преимущества данной программы:

- наличие визуальной среды разработки;
- межплатформенная поддержка;
- модульная система компонентов;
- понятный и простой интерфейс.



Малая часть игр, созданных в Unity:

- Cuphead

- Monument Valley 2
- Rick and Morty: VR
- Inside
- Ori and the Blind Forest
- Hearthstone
- Cities: Skylines
- Pokemon GO
- Temple Run
- Rust

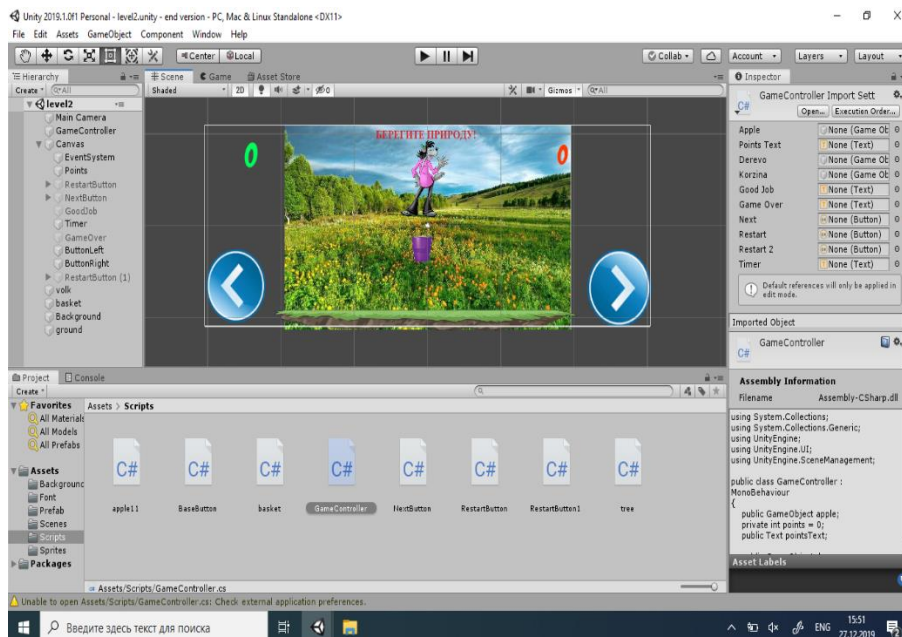
Концепция моей игры «Берегите природу»

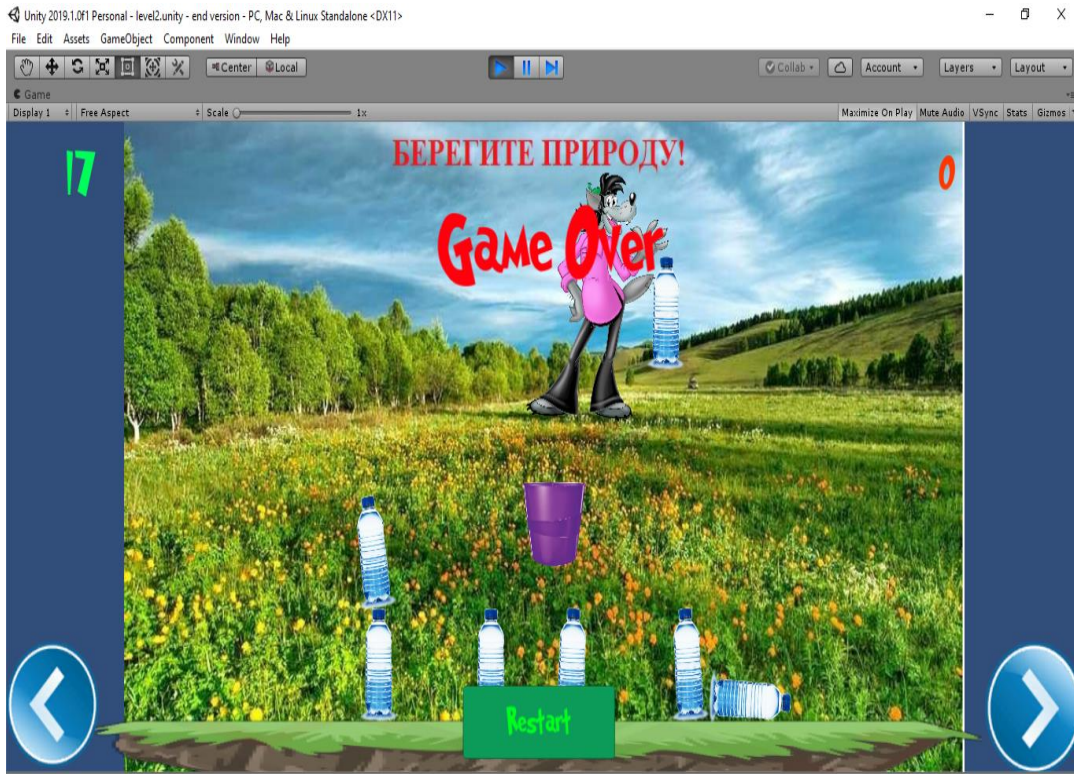
Социально – ориентированная игра: знакомство с экологической культурой, в процессе игры игрок должен понять что нельзя загрязнять окружающую среду;

Сценарий первой игры: игроку нужно будет собрать как можно больше мусора (бутылок), которые выкидывает отрицательный персонаж (в нашем случае Волк из мультфильма «Ну, погоди»), в корзину для мусора.

Реализация:

- персонаж, движение персонажа;
- накопление бутылок;
- ведение счета;
- отсчет времени.





```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class GameController : MonoBehaviour
{
    public GameObject apple;
    private int points = 0;
    public Text pointsText;
    public GameObject derevo;
    public GameObject korzina;
    public Text GoodJob;
    public Text GameOver;
    public Button next;
    public Button restart;
    public Button restart2;
    public Text timer;
    float time = 30;
}

```

```

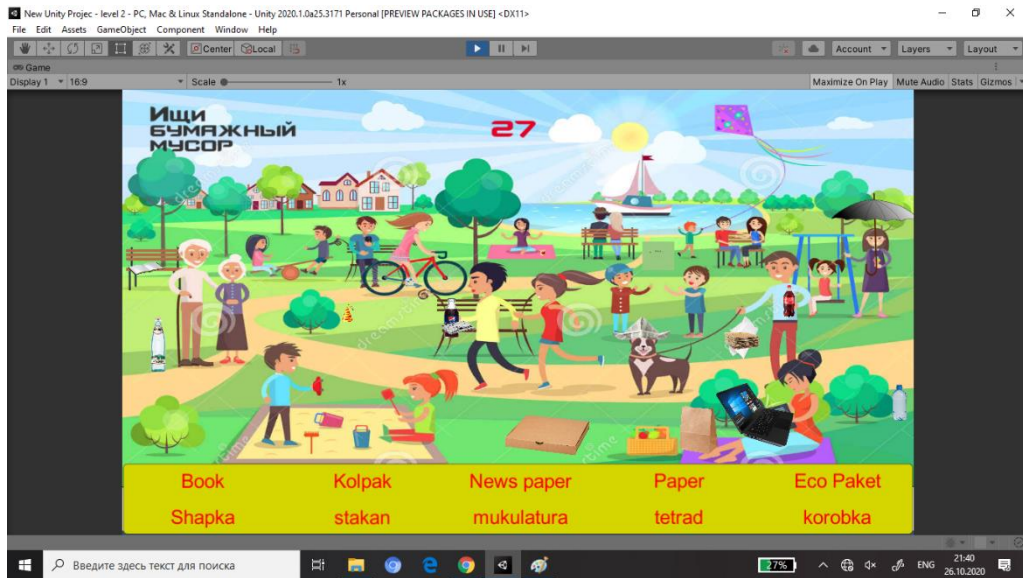
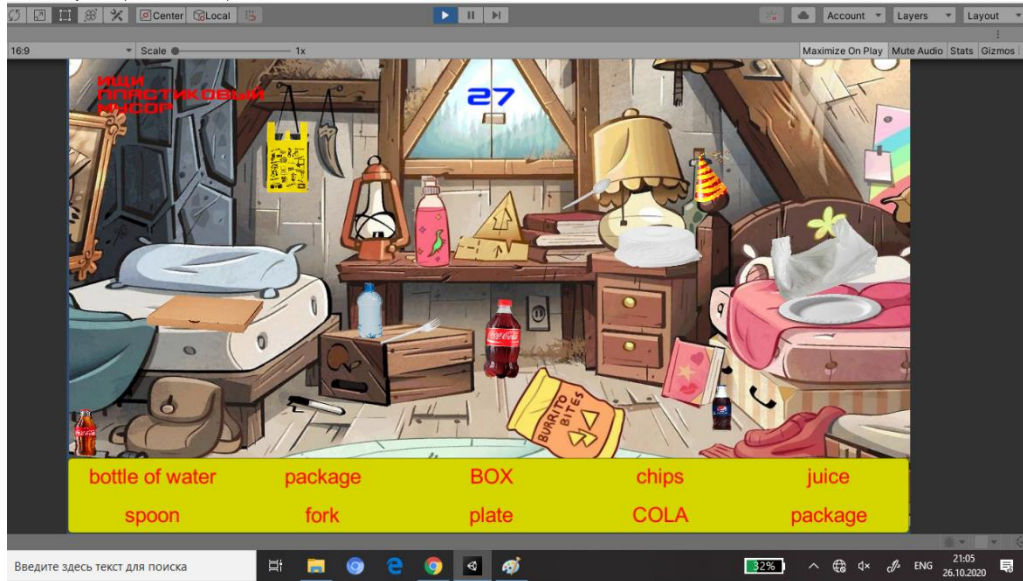
    public Stack<GameObject> Apples { get; set; } = new Stack
<GameObject>();
    private static GameController instance;
    public static GameController Instance
    {
        get
        {
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class tree : MonoBehaviour
{
    private Vector2 startPosition;
    private Rigidbody2D rb;
    public float speed;
    public float distance;
    private bool isMovingLeft = true;
    void Start()
    {
        startPosition = transform.position;
        rb = GetComponent<Rigidbody2D>();
        InvokeRepeating("RandomizeRotation", 0, 1);
    }
    private void RandomizeRotation()
    {
        var rand = Random.Range(0, 11);
        if (rand < 5)
        {
            isMovingLeft = true;
        }
        else
        {
using System.Collections;
using System.Collections.Generic;
using UnityEngine
public class basket : MonoBehaviour
{
    private Rigidbody2D rb;

```

```

public float speed;
public int directionInput;
public bool facingRight = true;
void Start()
{
    rb = GetComponent<Rigidbody2D>();
}
void Update()
{
    if ((directionInput < 0) && (facingRight))
    {
        Flip();
    }
    if ((directionInput > 0) && (!facingRight))
    {
        Flip();
    }
}
void FixedUpdate()
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class apple11 : MonoBehaviour
{
    void Update()
    {
        if (transform.position.y < -8)
        {
            gameObject.SetActive(false);
            GameController.Instance.Apples.Push(gameObject);
        }
    }
}
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class gameText : MonoBehaviour
{
    public GameObject objText;
    void Start()
    {
    }
}

```



```

// Update is called once per frame
void OnMouseDown()
{
    if(gameObject.name == objText.name)
    {
        Destroy(gameObject);
        Destroy(objText);
    }
}
}
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class TimerScripts : MonoBehaviour
{
    public int time;
    public Text timerText;
    public Image gameOver;
    public Button restart;
    // Start is called before the first frame update
    void Start()
    {
        InvokeRepeating("Timer", 0, 1); //повторение метода Timer
    //Update is called once per frame
    void Update()
    {
        timerText.text = "Timer: " + time.ToString(); //Вывод значения
времени
    }
    public void Timer()
        if(time == 0) //проверка закончилось время таймера
        {
            Time.timeScale = 0;
            gameOver.gameObject.SetActive(true);
            restart.gameObject.SetActive(true);
using System.Collections;
using System.Collections.Generic;

```

```

using UnityEngine;
using UnityEngine.UI;
public class GameController : MonoBehaviour
{
    public int time;
    public Text timerText;
    public Button restart;
    public Button next;
    public int totalClicks = 10;
    public KeyCode mouseClick;
    void Start()
    {
        InvokeRepeating("Timer", 0, 1);
    }
    void Update()
    {
        if (Input.GetKeyDown(mouseClick))
        {
            totalClicks--;
        }
        else if( totalClicks == 0 && GameObject.FindWithTag("Stuff") ==
null)
        {
            Win();
        }
    }
}

```

Результаты апробации:

- распространили игру среди учеников 4-х классов (всего скачали – 27)
- провели анкетирование:
 1. Понравилась игра/не понравилась (15/12)
 2. Смогли ли вы пройти все уровни? (10 да/17нет)
 3. Какая игра понравилась больше? (1игра-6, 2игра-21)
 4. Нужно ли увеличить время прохождения игры? (2-да, 25-нет)
 5. Нужно ли добавить фоновую музыку (23да/4нет)

В целом игры понравилась, но надо сделать больше уровней, чтобы игра была более азартной, и думаю что нужны «бонусы» (остановка времени, премиальные очки), а также реализовать третью игру «Сортировка мусора по уровню опасности».

Выводы:

При реализации проекта были решены следующие задачи:

- 1) изучена литература;
 - 2) изучена программа по разработке игр Unity;
 - 3) отобрано содержание игры;
 - 4) разработана программы игры (персонаж, правила, дизайн);
- изучен игровой движок Unity и были приобретены необходимые знания и умения, а именно (создание сцен, создание и написание скриптов, настройка объектов, создание UI, компиляция проекта).
- 5) апробирована игра.

Таким образом, на платформе «Unity» были разработаны игры «Берегите природу» и прошли успешную апробацию среди детей младшего школьного возраста.

СПИСОК ЛИТЕРАТУРЫ

1. IT-школа Алтан [Электронный ресурс] <https://altanschool.ru/>
2. Википедия-свободная энциклопедия [Электронный ресурс] <https://ru.wikipedia>
3. Unity [Электронный ресурс] <https://unity.com/ru/madewith>
- 4.Новости якт[Электронный ресурс] <https://news.ykt.ru/article/109806>
- 5.рэш [Электронный ресурс] <https://resh.edu.ru/subject/lesson/5504/conspect/298880/>
6. Википедия-свободная энциклопедия [Электронный ресурс] <https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D1%85%D0%BE%D0%B4%D1%8B>

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОИСКА РЕПЕТИТОРОВ

Чабыкин Олег

Физико-Технический лицей имени В.П. Ларионова, 9 класс

Проблемы:

- У школьников часто возникает потребность найти репетитора, а у репетиторов найти учеников.
- Чаще всего быстро найти репетитора удается не всегда

- Нету централизованного списка репетиторов, в котором был бы рейтинг и можно было бы писать отзывы и оставлять оценку услуг.
- Необходимо приложение под самую распространенную платформу, чтобы им могли пользоваться люди любого возраста.

Гипотеза: Тем, кто сталкивается с такими проблемами, будет выгодно пользоваться этим приложением. Лично я сталкивался с такой проблемой, что не мог быстро и удобно найти репетитора. Я провел поиск приложения в google play для того, чтобы найти репетитора, но в итоге так и не нашел продукта, ориентированного под локальный или всероссийский рынок.

Цель: разработать мобильное приложение под Android, чтобы помочь школьникам и репетиторам эффективно находить друг друга.

Задачи:

- Разработать серверную часть(backend).
- Реализовать взаимодействие между приложением и серверной частью.
- Ввести рейтинговую систему, систему отзывов.
- Дойти до стадии MVP.

Ход работы:

1. Анализ рынка: Я проанализировал рынок и пришел к выводу, что первоначально стоит ориентироваться под локальный рынок.
2. Проектирование архитектуры:
 - Я выбрал rest-архитектуру для взаимодействий клиента и сервера.
 - Android приложение выступает в качестве клиента, отправляющего запросы на сервер.
 - Серверная часть отвечает на запросы клиента json форматом.
3. Проектирование интерфейса (UI).
4. Разработка.

Технологии:

- Для серверной части я решил использовать язык Java и веб-фреймворк Spring, так как я считаю, что Java является одним из лучших выборов для разработки.
- В качестве базы данных я выбрал PostgreSQL.
- Для того, чтобы отправлять http запросы из android-приложения я выбрал библиотеку retrofit2 и для сериализации json библиотеку Gson.

- Инструменты: Для серверной разработки я использовал IntelliJ Idea, для android разработки использовалась Android Studio.

Серверная часть:

- Backend отвечает на http запросы андроид приложения в формате json.
- Сервер реализован в rest архитектуре.
- Взаимодействует с базой данных, в которой хранятся данные пользователей.
- Неавторизованные пользователи не смогут получить информацию о других пользователях.
- Сервер имеет хорошую производительность.

Чтобы веб сервер был доступен в интернете, я воспользовался сервисом heroku. Теперь серверная часть работает на серверах heroku.

В итоге были реализованы все функции, которые планировалось реализовать:

- Темная тема
- Рейтинг
- Система отзывов
- Отправка заявки
- Категории предметов
- Очки
- Просмотр отзывов
- Возможность написать о себе

Пользователь регистрируется, выбирает свою роль(репетитор,ученик), ищет репетитора по категориям, смотрит отзывы, выбирает репетитора, отправляет заявку со своим номером. Учитель связывается по этому номеру с отправителем заявки.

Заключение:

- Разработана серверная часть, подходящая под все требования.
- Было реализовано взаимодействие приложения с сервером, с помощью подходящей библиотеки.
- Разработано функциональное андроид приложение, позволяющее находить репетиторов.
- Приложение доступно в Play Market.

СОЗДАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ СИНХРОНИЗАЦИИ ДАННЫХ С СЕРВЕРОМ

Пуртов Глеб

*МАОУ «Красноярская университетская гимназия 1 – Универс»,
г. Красноярск, 10 класс*

Руководитель: Вахитова Е.Ю., учитель информатики

Сейчас существует множество облачных сервисов, которые позволяют как безопасно сохранять важные файлы, так и синхронизировать свою фотопленку. Как правило такие сервисы не бесплатны или имеют определенные ограничения. К тому же нельзя гарантировать сохранность и безопасность данных, ведь физически они хранятся на чужом компьютере. Все эти вопросы решаются с помощью данного приложения. Оно позволяет использовать любую флешку и домашний роутер в качестве сервера, на котором можно сохранять фотографии прямо с мобильного устройства. К тому же это не требует материальных затрат и не имеет ограничений на объем хранимых данных.

Приложение актуально для любых пользователей смартфонов и других мобильных устройств и позволяет без затрат на оборудование и облачные сервисы сохранять данные. Для его функционирования нужна рабочий USB-флеш накопитель и домашний роутер.

Цель работы: Создание мобильного приложения для удаленной синхронизации данных с сервером.

Задачи:

1. Составить схему работы приложения в виде алгоритма исхемы.
2. Изучить теоретические основы создания мобильных приложений.
3. Выбрать подходящую среду для разработки приложения.
4. Изучить способ взаимодействия приложения с протоколом FTP.
5. Изучить аналоги, существующие на рынке.
6. Написать программный код для реализации функционала приложения.
7. Создать дизайн приложения.
8. Апробация приложения, тестирование на стабильность.

Оборудование, методы:

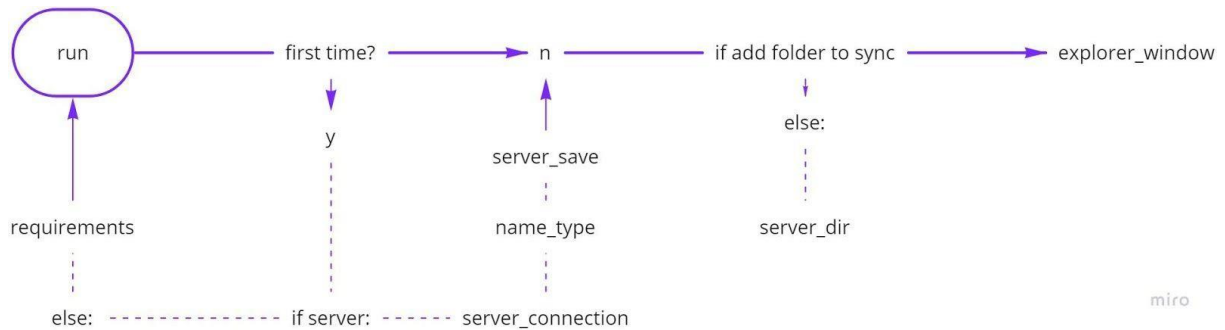
Работа проводилась в среде разработки Visual Studio Code. Для контроля версий использовался Git и сервис GitHub. Проект реализован на

языке программирования Python версии 3.7.6. В ходе работы были использованы библиотеки os - работа с директориями, pillow - работа с изображениями, ftplib - работа с FTP, PyQt5 - реализация графической оболочки, configparser - взаимодействие с файлами конфигурации sys - необходимый компонент для работы Qt, requests - проверка доступности сервера.

Ход работы

Составление схемы работы приложения:

Перед началом работы было реализовано графическое представление примерного алгоритма работы приложения.



Написание программного кода.

За интерфейс отвечает файл «main.py», в котором загружается файл «interface.py» из папки «ui». При первом запуске пользователь вводит данные сервера, логин и пароль. Они сохраняются в файл «config.ini» методом «config_rewrite»:

```

def config_rewrite(self, ip, username, password):
    config.set('server', 'ip', ip)
    config.set('client', 'username', username)
    config.set('client', 'password', password)
    with open('config.ini', 'w') as configfile:
        config.write(configfile)
  
```

После этого пользователь выбирает папку, которую требуется синхронизировать (еще в разработке) и помощью метода «gallery_load» ему выводятся все фотографии из данной директории:

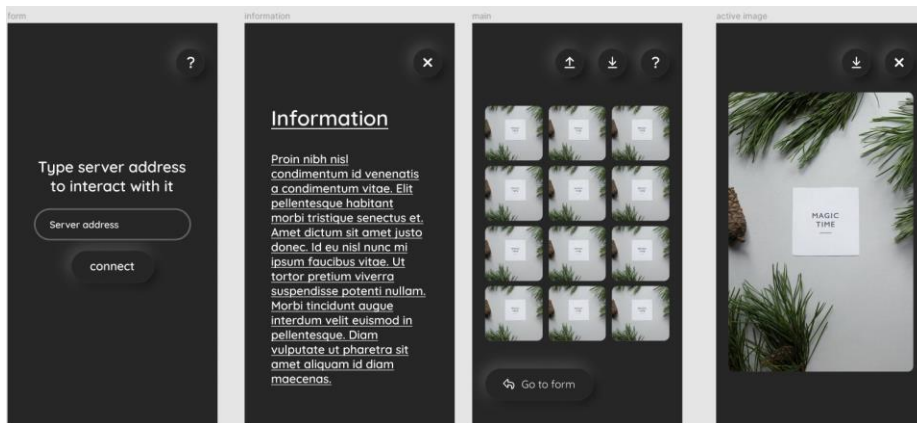
```
def upload_all(self):
    uploaded = False
    current_dir = os.listdir(upload_dir)
    desired_dir = '/' + server_dir + '/'
    desired_dir_files = self.ftp.nlst()
    self.ftp.cwd('/') + server_dir + '/'
    if current_dir:
        for uploading in current_dir:
            if desired_dir + uploading in desired_dir_files:
                continue
            send = os.path.join(upload_dir, uploading)
            try:
                self.ftp.storbinary(f'STOR {uploading}', open(send, 'rb'))
                uploaded = True
                compress(send)
                print('success')
            except ftplib.error_perm as codes:
                error_code = str(codes).split(None, 1)
                if error_code[0] == '550':
                    print(
                        error_code[1], "you don't have permission to upload here")
                    uploaded = False
            if not uploaded:
                print('all files are already on server')
    else:
        print('gallery is clear')
```

После нажатия на любую из картинок из галереи пользователь попадает на предпросмотр фотографии – отсюда, в случае если фотография находится на сервере, он может скачать несжатую версию, которая заменит сжатую или загрузить ее на сервер.

В настройках пользователь может изменить данные сервера и входа, расположение галереи.

Создание дизайна приложения:

Дизайн создавался в бесплатном сервисе Figma. Был выбран стиль «неоморфизм» за его ненавязчивость и приятный внешний вид. На скриншотах демонстрируются все рабочие экраны приложения. Экран авторизации, информации, общий вид галереи и отдельного фото.



Выводы:

- В результате данного проекта были изучены библиотеки PyQt, Pillow, ftplib, configparser.
- Была изучена среда разработки интерфейсов Figma.
- Была изучена среда разработки интерфейсов Qt Designer.
- Были созданы основные необходимые классы — App, FTP.
- Были реализованы основные механики приложения — загрузка на сервер, загрузка с сервера, компрессия (сжатие) изображений.

Заключение:

Приложение все еще подлежит небольшим улучшениям, но на текущем этапе было реализовано большинство необходимых механик. В дальнейшем данное приложение позволит быстро, безопасно и бесплатно сохранять данные для любых пользователей.