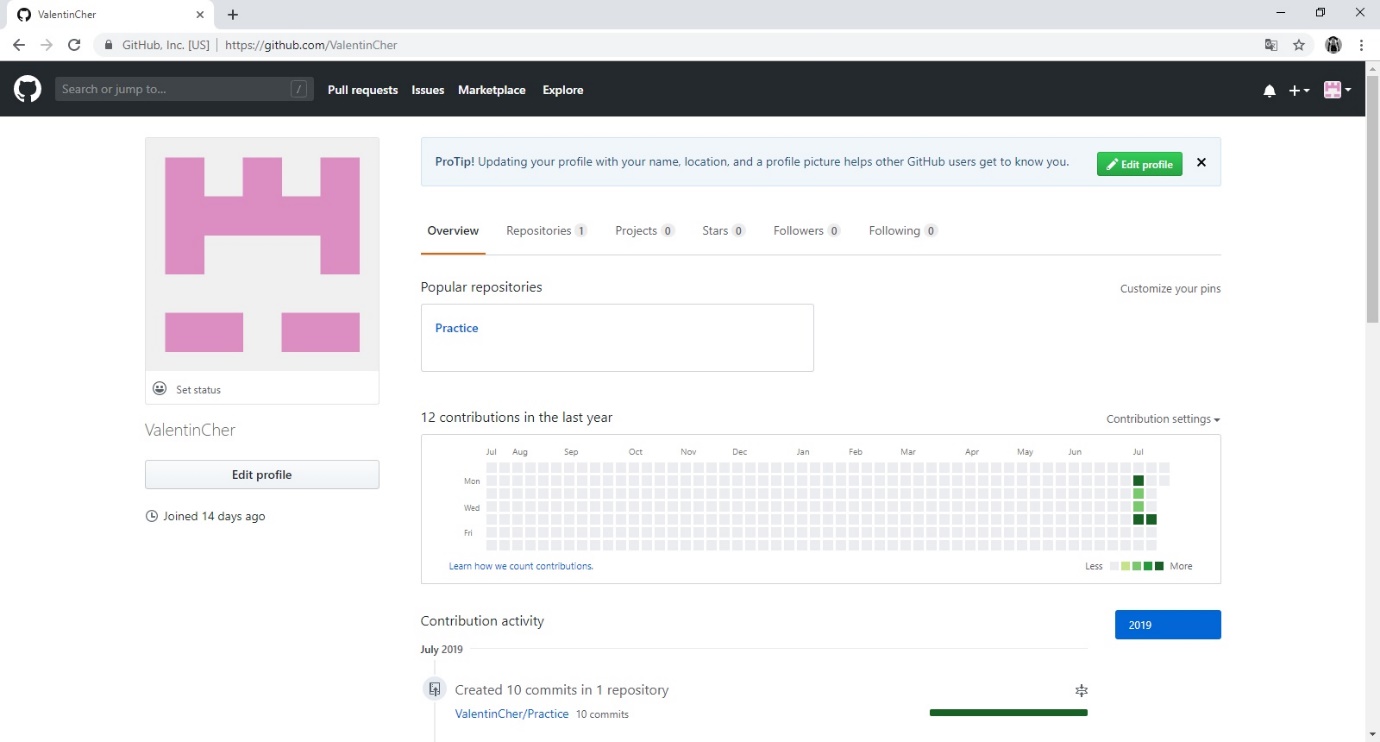
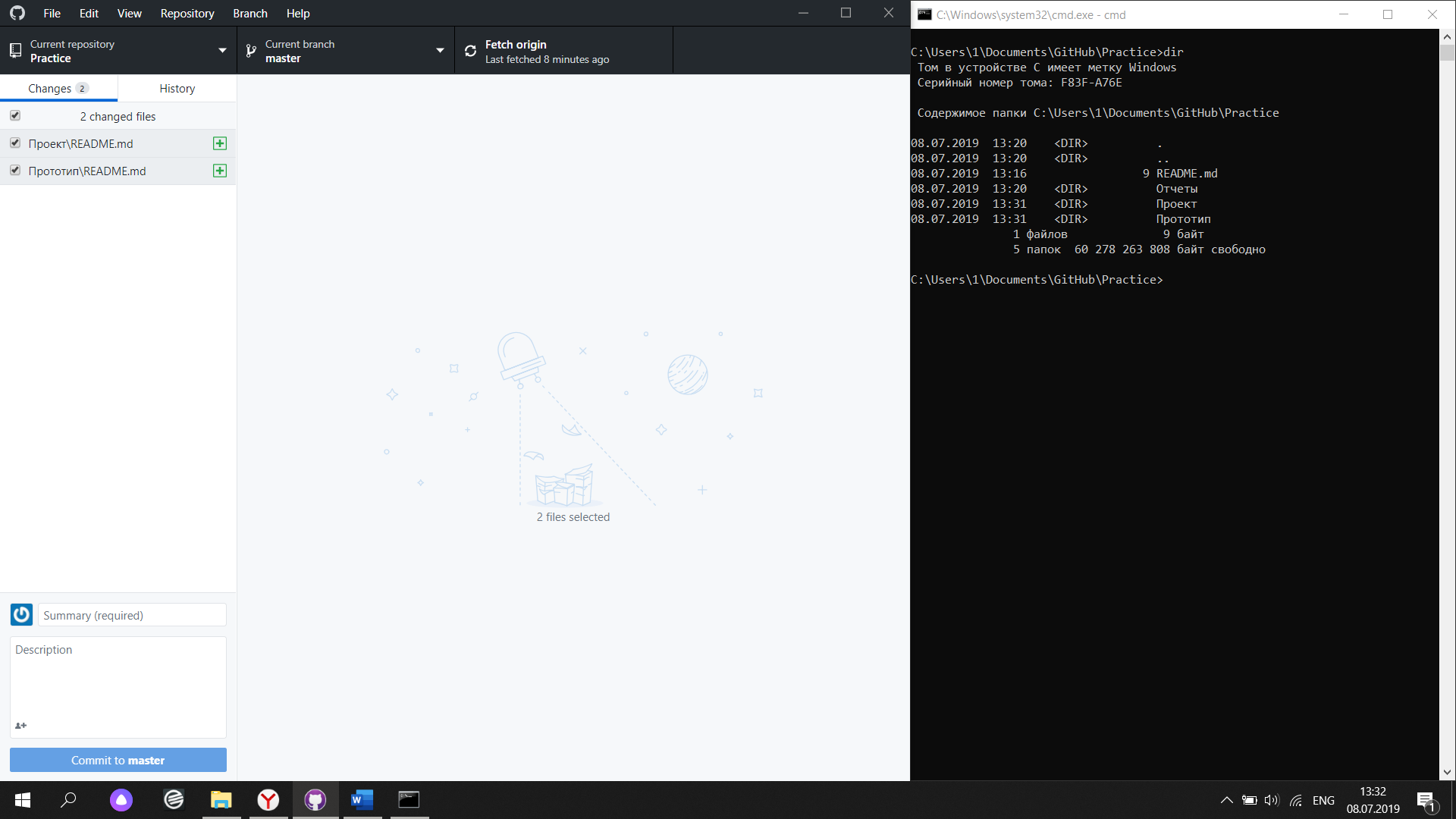
**День 1**

****



**Что такое Git и зачем его использовать?**

Git — это система управления версиями. У Git две основных задачи: первая —  хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

Репозиторий Git — это место, где хранится ваш код и вся информация о его изменениях. Репозитории могут находиться у вас на компьютере, на компьютерах ваших коллег и на удалённом сервере.

## КАК ПОЛЬЗОВАТЬСЯ GIT?

Кроме установки, вам нужно указать правильный адрес электронной почты и имя пользователя для доступа к серверу Git, например, на GitHub.

Обычно, структура проекта в Git будет зависеть от масштаба и сложности вашей программы. Но для начала мы будем использовать проект, состоящий только из одной ветви. Каждый проект содержит одну ветку по умолчанию, она называется master.

### СОЗДАНИЕ ПРОЕКТА

Когда настройка git завершена перейдем к вашему проекту. В самом начале вам достаточно создать папку для файлов проекта. Если вы собираетесь работать над несколькими проектами, создайте папку git в вашем домашнем каталоге, а уже туда поместите папки ваших проектов:

 mkdir -p ~/git/testing ; cd ~/git/testing

Эта команда создаст нужную структуру папок и переводит текущий каталог в только что созданный. Теперь создадим первый файл нашего проекта:

 touch file

Проект готов, но система контроля версий git еще не знает об этом.

### НАСТРОЙКА ПРОЕКТА В GIT

Перед тем как git начнет отслеживать изменения, нужно подготовить все необходимые конфигурационные файлы. Сначала инициализируем пустой репозиторий в нашей папке:

 git init

После того как репозиторий будет создан, вам нужно добавить свои файлы в него. Каждый файл нужно добавлять отдельно или сказать утилите, что необходимо добавить все файлы явно. Пока вы не добавите файл сам он не будет отслеживаться. Новые файлы в будущем тоже нужно добавлять, они не добавляются автоматически.  Сначала добавим текущую папку:

 git add .

Если все прошло хорошо, то команда ничего не выведет.

### ФИКСАЦИЯ ИЗМЕНЕНИЙ

Изменения тоже автоматически не отслеживаются. Фиксация изменений выполняется с помощью команды commit. Вам нужно указать что было изменено с помощью небольшого комментария, буквально в несколько предложений. Хорошая практика выполнять фиксацию перед каждым серьезным изменением.

Таким образом, вы будете хранить все версии проекта, от самой первой и до текущей, а также сможете знать что, когда и где было изменено. Чтобы создать свой первый коммит выполните:

 git commit -m "Initial Commit" -a

Команде необходимо передать два параметра, первый - это -m, ваш комментарий, второй -a, означает, что нужно применить действие ко всем измененным файлам. Для первого раза используется этот параметр, но обычно вам нужно указать измененные файлы или каталоги. Например, можно делать так:

 git commit -m "Changed file" file

### ОТПРАВКА ИЗМЕНЕНИЙ

До этого момента мы делали все в локальном репозитории. Вы можете использовать git локально, если нужен только контроль версий, но иногда нужно обменяться информацией с другими разработчиками и отправить данные в удаленный репозиторий.

Сначала нужно добавить удаленный репозиторий с помощью команды remote. Для этого нужно передать ей URL:

 git remote add origin https://github.com/Seriyyy95/testing.git

Затем можно посмотреть список удаленных репозиториев:

 git remote -v

Вы можете использовать не только github сервера, но и любые другие. Теперь для отправки ваших изменений используйте такую команду:

 git push origin master

Команда push указывает, что нужно отправить данные в удаленный репозиторий, origin - наш настроенный репозиторий, а master - ветвь.

### УПРАВЛЕНИЕ ВЕТВЯМИ

Для простых проектов достаточно одной ветви. Но если проект большой и он имеет несколько версий, в том числе тестовую, то может понадобиться создать для каждой из них отдельную ветвь. Сначала смотрим доступные ветви:

git branch -a

Опция -a указывает что нужно вывести все ветви, даже не синхронизированные. Звездочка указывает на активную ветвь. Теперь создадим ветвь для разработки с помощью команды checkout:

 git checkout -b develop

Переключаться между ветвями можно тоже с помощью той же команды:

 git checkout master  
$ git checkout develop

Теперь создадим еще один файл:

 touch develop

И добавим его в нашу новую ветвь develop:

 git add develop

Сделаем коммит для внесенных изменений:

 git commit -m "develop file" develop

Дальше проверим существует ли этот файл в основной ветке master или только в дополнительной. Смотрим текущую ветку:

 git branch  
$ ls

Затем переключаемся на ветку master и снова смотрим:

 git checkout master  
$ git branch  
$ ls

Здесь файла нет, так и должно быть. В git есть такая полезная вещь, как слияние. С помощью нее вы можете объединить две ветви. Например, переместить код из рабочей ветки в стабильную. Для этого достаточно выполнить команду merge:

 git merge develop --no-ff

Перед тем как будет выполнено слияние вам нужно ввести комментарий, зачем это нужно. Затем если вы еще раз выполните ls, то увидите, что здесь уже есть нужный файл. Наши примеры git подошли к концу.

**День 2**

**Стек протоколов Bluetooth**

Этот раздел посвящен стандартам и протоколам, из которых состоит стек протоколов Bluetooth (рис. 17.1). Эти протоколы и стандарты исчерпывающе характеризуют Bluetooth: те виды данных, для передачи которых предназначен Bluetooth, количество устройств, между которыми можно одновременно установить связь, время задержки при передаче и т. д.

[](http://src-code.net/img/Android_programming_for_mobile_image479.jpg)

Рис. 17.1. Стек протоколов Bluetooth в Android

Bluetooth появился как особый род сетевых взаимодействий, поскольку он представляет собой персональную, или личную, сеть. Такие сети также называются аббревиатурой PAN, или пикосетью. Bluetooth предназначен для соединения до восьми устройств и для передачи данных с максимальной скоростью до 3 Мбит/с. Соединенные устройства должны располагаться близко друг к другу: в пределах 10 м. Bluetooth потребляет очень малое количество энергии, исчисляемое милливаттами. Это означает, что даже самой маленькой батареи надолго хватает для такой связи. Наушники Bluetooth с крошечной, невесомой батареей могут часами обеспечивать голосовую связь – примерно столько же выдерживает гораздо более крупная батарея мобильного телефона, поскольку мобильный радиосигнал с телефона должен достигать значительно более отдаленной антенны.

К приборам, с которыми полезно использовать Bluetooth, относятся устройства с низкой и средней скоростью передачи данных – в частности, клавиатуры, мыши, планшеты, принтеры, микрофоны, наушники (в том числе с микрофонами), а также мобильные и персональные компьютерные устройства, если с их периферийными компонентами может потребоваться связь. Bluetooth также позволяет устанавливать соединения между ПК и мобильными телефонами.

**Протоколы, специфичные для Bluetooth, и заимствованные протоколы**

Говоря о стеке протоколов Bluetooth, важно разделять эти протоколы на специфичные для Bluetooth и заимствованные, то есть такие, которые работают «поверх» Bluetooth. Вместе все эти протоколы – как относящиеся непосредственно к Bluetooth, так и заимствованные – могут быть очень сложны. Но если на минуту абстрагироваться от того, что над Bluetooth находятся такие сложные протоколы, как OBEX и TCP/IP, то все становится гораздо понятнее. Поэтому мы начнем с более низких уровней Bluetooth и покажем, как эти уровни помогают «оформить» нашу работу с Bluetooth.

Еще одна важная абстракция при работе с Bluetooth сводится к тому, что эта технология заменяет порты последовательного ввода-вывода. Это означает, что нижние уровни Bluetooth эмулируют виртуальный набор последовательных кабелей между соединяемыми устройствами и позволяют вам управлять этими кабелями. Именно с таким протоколом Bluetooth мы и будем работать. Таким образом, мы сможем пользоваться для считывания и записи данных простыми классами ввода-вывода Java. іо: InputStream и OutputStream.

**Bluez: реализация Bluetooth для Linux**

Мобильное устройство может соединяться по Bluetooth с какими угодно другими устройствами – чего не скажешь о периферийных устройствах, которые могут подключаться только к компьютеру или мобильному устройству. Это означает, что на мобильном устройстве нам потребуется практически полная реализация Bluetooth и всех заимствованных протоколов, а также пользовательский интерфейс, обеспечивающий необходимые взаимодействия для установления и использования соединений, а также для работы в приложениях, связывающихся по Bluetooth.

В Android используется стек Bluez Bluetooth – наиболее распространенный стек для работы с Bluetooth, применяемый в Linux. Он пришел на смену проекту, называвшемуся Open ВТ. Информация о Bluez содержится на сайте проекта Bluez – http://www.bluez.org.

Bluez, разработанный в Qualcomm, вошел в состав ядра Linux. Проект начался в 2001 году и до сих пор активен, хорошо поддерживается. Кроме того, Bluez – это стабильная реализация, обладающая хорошей совместимостью, – еще одна причина, по которой стоит использовать Linux в операционных системах для мобильных устройств.

**Использование Bluetooth в приложениях Android**

Под использованием Bluetooth в Android понимается применение классов, специально разработанных для того, чтобы инкапсулировать принцип, по которому Bluetooth работает в операционной системе Android. Стек Bluez предоставляет методы для перечисления устройств, слушания соединений и использования соединений. В пакете Java. іо предоставляются классы для считывания и записи данных; а классы Handler и Message обеспечивают возможности построения связей (мостов) между пользовательским интерфейсом, с одной стороны, и потоками, управляющими вводом и выводом в Bluetooth, – с другой. Рассмотрим код и примеры использования этих классов.

Компиляция и запуск кода помогают понять, что могут делать классы Bluetooth в приложениях Android, предназначенных для установления связи между устройствами, которые расположены вблизи друг от друга.

Начнем испытание программы, использующей Bluetooth, с самого простого: попробуем соединить мобильный телефон с персональным компьютером. Затем нам понадобится программа, которая наблюдала бы за той информацией, которую получает ПК по Bluetooth, и следила, какая именно посланная вами информация дошла до ПК. В данном случае применяется утилита Linux, называемая hcidump.

Запустите программу в режиме отладки и установите точки останова в частях приложения, которые занимаются открытием и принятием соединения. Можно создать соединение со своего ПК (в Linux для этого используется апплет Blueman) или же из приложения. После создания соединения запустите на терминале программу hcidump и убедитесь, что информация, которую вы ввели, попала на ПК. Пользуйтесь перечисленными ниже флагами, чтобы отображать только то содержимое, которое поступает по Bluetooth:

**Аналоги мобильных игр по Bluetooth:**

**Для iPhone:**

**“Крестики Нолики (Tic Tac Toe)”**

Старомодный Tic-Tac-Toe теперь доступна на вашем телефоне.

Вы можете играть с Android, с другим игроком на своем телефоне или через Bluetooth.

Выберите размер платы (3х3 мин, макс 15х15), установить, сколько знаков игрока определяют победителя и начать игру.

Игра крестики нолики на iPhone особенности:

- 1 Игрок против iPhone или игры крестики нолики на 2 игрока с тем же устройством

- Навык уровней: легкий, средний, жесткий или эксперт

- Сбрасываемые статистика и оценка отслеживания

- Несколько игры крестики нолики темы чтобы выбрать из

- Автоматическое сохранение, когда вы получаете прерывание по телефону или выйти из приложения

**“Судоку - Игра-головоломка”**

Логические головоломки, пазлы

Судоку – ваша любимая классическая игра-головоломка. Оцените удобство судоку на мобильном устройстве, это отличная альтернатива бумаге и карандашу! Игра понравится как новичкам, так и опытным игрокам. Тренируйте память, развивайте логическое мышление и получайте удовольствие!

Мы добавили несколько уровней сложности – от легкого до сложного, которые подойдут каждому, а также множество функций для упрощения игры, но оставили выбор за вами, использовать их или нет. При этом каждое судоку имеет только одно решение.

Функции игры

· Ежедневные испытания. Решайте судоку каждый день и получайте уникальные награды

· Заметки. Добавляйте предполагаемые варианты решения в ячейку, чтобы не держать их в голове

· Подсказки. Воспользуйтесь подсказкой, если запутались или не знаете правильного решения

· Выберите одну из трех тем для игры в Судоку с комфортом, даже в темноте!

· Проверка ошибок. Включите функцию автоматической проверки, чтобы видеть свои ошибки сразу

· Выделение повторяющихся цифр. Воспользуйтесь этой функцией, чтобы избежать повторов в строке, столбце и блоке

· Стикеры для iMessage. Дарите друг другу позитивные эмоции, используя весёлые стикеры с ниндзя!

**“Крестики-нолики 2 Онлайн”**

Основные особенности игры:

ИГРА С ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ

В Крестики-нолики 2 можно играть с ИИ, выбрав для себя подходящий уровень сложности.

Легкий – уровень для детей!

Средний – Ваши умения будут проверены!

Сложный – скорее всего Вы не одержите победу!

ИГРА ПО БЛЮТУЗУ

Играйте по блютузу с друзьями, коллегами или просто знакомыми!

ОНЛАЙН ИГРА

Сразитесь с соперниками по ИНТЕРНЕТУ!

**Для Android:**

**“Виселица”** - игра на ваше устройство Android.

Игра поддерживает одно-и многопользовательские.

Виселица это простая игра. Вы должны угадать слово, предлагая отдельные буквы.

Прекрасная игра, чтобы играть с детьми!

Одиночном режиме обеспечивает три уровня сложности. Совершенствуйте свои навыки!

Подключение к другому устройству через Bluetooth Android и играть с друзьями.

Прекрасные возможности:

• Bluetooth многопользовательских

• статистику ваших игр

• Встроенный чата в сетевом

• Блестящая графика и звуковые эффекты

• английского и немецкого языков в одиночном режиме

\* Простой многопользовательской игры Bluetooth для Android \*

**“Two guys & Zombies”** - красивый мультяшный 2д зомби шутер на двоих. Если ищите с другом игры по bluetooth для того, чтобы сыграть в них вдвоем, то это то, что вам нужно.   
Здесь вы играете за ковбоя и полицейского, которые оказываются посреди города в окружении зомби. Ничего не остается делать, как сражаться с этими тварями до последней капли крови.   
  
Особенности:  
• Игра на двоих по bluetooth  
• Прокачка героя  
• Строительство баррикад, турелей и т.д.  
• Множество разновидностей зомби  
• Множество интересных локаций  
• Различные оружия от пистолета до гранатомета  
• Приятная графика и звуковое сопровождение

Цель в самой игре продержаться как можно больше волн зомби, для того, чтобы заработать алмазы. За алмазы вы можете покупать способности для вашего героя. С каждой новой способностью вы сможете продержаться ещё дольше. Например, способность "Удобная обувь" позволит герою двигаться с более быстрой скоростью, что пригодиться в тактическом отступлении в опасной ситуации. Или, например, способность "Бронежилет" позволит герою переносить больше ударов от зомби.  
  
Гораздо проще играть в Two guys & Zombies на двоих, так как в игре часто зомби будут наступать с двух сторон и всегда приятно, когда вашу спину прикрывает ваш товарищ. Ещё более удобно, когда ваш друг строит баррикады, а вы защищаете его, отстреливаясь от зомби.

**“Шахматы”** - приложение, в котором вы можете играть в шахматы в дороге, через Bluetooth, у неё есть замечательный интерфейс пользователя, и хорошая функциональность  
  
 \* Вы можете играть в шахматы с друзьями или лицом к лицу или через Bluetooth.  
 \* Опция для игры с компьютером доступно  
 \* Замечательный интерфейс пользователя  
 \* Если вы играете с помощью Bluetooth, если требуется вы можете общаться с вашим другом, а также  
  
 Основные характеристики:  
      - Играть с помощью Bluetooth  
      - Чат с помощью Bluetooth  
      - Играть с Компьютера  
      - Замечательный интерфейс пользователя, умный AI

**“Точки” -** логическая настольная игра на клеточной бумаге. Теперь в роли листика и ручки – ваш смартфон! Играй с соперниками со всего мира благодаря онлайн мультиплееру! Тренируйся в играх с Андроидом!  
Цель игры — окружить как можно больше точек противника. Для этого соперники по очереди ставят точки на пересечении линий листа в клетку, каждый своим цветом. Окружение должно быть построено так, чтобы расстояние между точками составляло не более одной клетки — по горизонтали, вертикали или по диагонали. Партия заканчивается, когда не осталось свободных мест, по взаимному согласию игроков, либо когда один из игроков сдался.  
Точки — игра с красивой графикой в тетрадочном стиле, что придаёт игре оригинальность и незабываемую атмосферу!

**“Морской бой 2”** - всеми любимая с детства настольная игра с новыми возможностями и расширенным арсеналом! Миллионы людей по всему миру играют в эту игру. В твоём распоряжении корабли, самолёты, подводные лодки, мины, радары и не только. Расставляй на поле боя свои корабли, наноси удары по полю врага, используй арсенал, стараясь потопить корабли противника. Построй свою стратегию игры и не оставляй соперникам шансов!

Сражайся с соперниками со всего мира через интернет в режиме реального времени! Участвуй в боях между платформами!

Морской бой — игра с красивой графикой в тетрадочном стиле и эффектами, что придаёт игре оригинальность и незабываемую атмосферу!

Особенности игры:

-Сражайся с соперниками со всего мира через интернет! Защищай свою платформу! Каждый игрок может повлиять на результат!

-Побеждай в боях, чтобы повысить свой ранг и звание от курсанта до адмирала!

-Выбирай подходящий уровень сложности и пробуй победить ИИ (искусственный интеллект). Зарабатывай очки и повышай звание!

-Устраивайте с друзьями, коллегами или просто знакомыми сражения по блютузу.

-Играйте с друзьями на одном телефоне (планшете), расставляйте по очереди корабли, выбирайте арсенал и в бой!

-Общайся с соперниками во время боя с помощью чата. Выбери себе грозное имя и флаг своего флота.

-Выбирай классический или расширенный режим игры. Изменяй арсенал по своему усмотрению.

**“Warlings”** - устройте Армагеддон врагам и всему, что их окружает, в этой новой версии пошаговой стратегии Warlings, в которую с удовольствием играет более 3 млн человек! Стройте планы и сражайтесь почти 30 видами оружия на 10 уникальных картах. Победите других игроков онлайн и получите звание Мастера. Также можно играть с друзьями!

ОСОБЕННОСТИ:

● Полностью разрушаемое поле боя

● 29 потрясающих видов оружия

● 10 качественных карт

● Бойцов можно экипировать по своему вкусу

● Играйте онлайн или по Bluetooth

● Рейтинг игр онлайн

● Режим «хотсит»"

**Сравнение данных приложений представлено в таблице 1:**

**Таблица 1:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название игры: | Рейтинг: | Типы устройств: | Размер: | Стадия проекта: | Платные услуги: | Количество установок: | Ссылки: |
| Bluetooth Tic-Tac-Toe | 4,5 | **Android** 2.2 и выше | 1,9M | Завершен | - | 500+ | https://play.google.com/store/apps/details?id=com.SAK.Game.Bluetooth.tictactoe |
| Морской бой 2 | 4,6 | Android  4.0 и выше | 33М | Обновляется | + | 600000+ | <https://play.google.com/store/apps/details?id=com.byril.seabattle2> |
| Tic Tac Toe Bluetooth | 3,6 | Android  2.3 и выше | 2,5М | Завершен | - | 10000+ | <https://play.google.com/store/apps/details?id=com.blogpentingstudio.tictactoebluetooth> |
| XXL Bluetooth Tic Tac Toe Free BT | 4,1 | Android  4.1 и выше | 2,7М | Завершен | - | 1000+ | <https://play.google.com/store/apps/details?id=com.thirteendollars.tictactoebl> |
| CHESS BLUETOOTH | 3,5 | Android  4.0 и выше | 4,2M | Завершен | + | 100000+ | <https://play.google.com/store/apps/details?id=innovationlabs.chess.com> |
| Warlings: Армагеддон | 4,4 | Android  4.0.3 и выше | 42M | Завершен | + | 10000000+ | <https://play.google.com/store/apps/details?id=com.warlingsarmageddon> |
| Точки Онлайн | 4 | Android  2.3.3 и выше | 10M | Завершен | - | 1000000+ | <https://play.google.com/store/apps/details?id=com.byril.dots> |
| Traffic Run! | 4,6 | iOS 9.0 и новее | 193,7МБ | Обновляется | + | 10000 | <https://apps.apple.com/ru/app/traffic-run/id1434400630> |
| Судоку - Игра-головоломка | 4,8 | iOS 10.0 и новее | 78,2МБ | Завершен | - | 200 | <https://apps.apple.com/ru/app/судоку-игра-головоломка> |
| 2 на 2 | 4,2 | iOS 9.0 и новее | 215,1МБ | Обновляется | + | 3000 | <https://apps.apple.com/ru/app/2-на-2> |
| Крестики Нолики | 3,8 | iOS 8.0 и новее | 84,4МБ | Обновляется | + | 500 | <https://apps.apple.com/ru/app/крестики-нолики> |
| Крестики-нолики 2 Онлайн | 4,4 | iOS 6.0 и новее  Android  2.3.3 и выше | 56,1МБ | Завершен | + | 1000 | <https://apps.apple.com/ru/app/крестики-нолики-2> |
| Дурак | 4,3 | Android  4.0.3 и выше | 48M | Завершен | - | 5000000+ | <https://play.google.com/store/apps/details?id=ru.appscraft.durak> |
| Мега Крестики-Нолики | 4,2 | iOS 8.0 и новее | 49,7МБ | Завершен | - | 200 | <https://apps.apple.com/ru/app/мега-крестики-нолик> |
| Hangman Free | 3,4 | Android  2.1 и выше | 2,7M | Завершен | - | 100000+ | <https://play.google.com/store/apps/details?id=com.thundrix.hangmanfree> |
| Hangman | 3,2 | Android  2.1 и выше | 2,7M | Завершен | + | 50+ | <https://play.google.com/store/apps/details?id=com.thundrix.hangman> |
| Tic Tac Toe | 5 | Android  2.1 и выше | 1,0M | Завершен | + | 5+ | <https://play.google.com/store/apps/details?id=com.thundrix.tictactoe> |
| Two guys & Zombies | 4,1 | Android  4.1 и выше | 43M | Обновляется | + | 100000+ | <https://play.google.com/store/apps/details?id=com.yad.twoguysandzombies> |
| ДУ BombSquad | 4,3 | Android  4.0 и выше | 1,3M | Завершен | - | 1000000+ | <https://play.google.com/store/apps/details?id=net.froemling.bsremote> |

При всей кажущейся простоте, популярная игра «крестики-нолики» требует повышенного внимания и отлично способствует развитию логики. В эту игру играют не только дети, есть любители поиграть в «крестики-нолики» и среди взрослых; дети учатся в этом случае основам логического мышления, а взрослые вспоминают его базу и получают разрядку на фоне напряжённых трудовых будней.   
  
Играть в «крестики-нолики» можно с товарищем или с виртуальным партнёром, выбрав для этого желаемый уровень сложности. Такое времяпровождение невозможно назвать бесцельным, ведь оно помогает расслабиться с одной стороны и настраивает на позитивное мышление с другой.   
  
  
  
**На чем писать мультиплатформенное desktop-приложение?**  
Сегодня авторы большинства приложений уже не могут позволить себе выпускаться под одну платформу. Early adopters сидят под маками, мейнстрим сидит под Win32, а гики и адепты open source предпочитают Linux. Каждая из этих аудиторий обладает уникальными свойствами, а поэтому важна для большинства проектов.  
  
Данная статья задумывалась как открытая попытка разобраться, на чем стоит писать мультиплатформенное desktop-приложение. Приглашаю высказать свое мнение людей с опытом создания таких приложений.  
  
Изначально определим критерии, по которым будем оценивать различные платформы. В первую очередь — это удобство пользователя. Уверен, что вы замечали общие черты у приложений, написанных под разные платформы. Во вторую — это интересы проекта. Моя задача — построить успешный бизнес, а не играться с различными инструментами.  
  
Обозначим рамки исследования. Мое приложение — небольшая утилита для пользователя-«чайника», которая качает файлы из интернета: минимум GUI, небольшой набор функциональности, использование внешних С++ библиотек.  
  
Ну что, начнем. Какие есть варианты? Я рассмотрю Java, C#, C++, Python. Буду рад, если вы расскажите о других альтернативах.   
Java   
Данный язык/среда изначально задумывались как нечто мультиплатформенное. На Java написано большое количество приложений, крупные проекты вроде Eclipse используют именно этот фреймворк.  
  
Большой минус Java с точки зрения пользователя — необходимость устанавливать фреймворк. Это безусловно сложнее установки Flash player'а и часто становится pain in the ass. И размер инсталлятора (я качал 80+ мб), и постоянная путаница в названиях (JRE, J2SE, JDK, JVM, ...) не играют на руку разработчикам приложений под Java.  
  
Learning curve имеет критическое значение для любого проекта, который ставит на вирусное продвижение. Если 30% ваших пользователей не смогут поставить Java-машину (лень качать, неудобный сайт, отложили «на потом», ...) — у ваших конкурентов будет фора в 30%. Эта огромная доля пользователей, потерять которых мой проект себе позволить не может.  
  
Все приложения, которыми мне доводилось пользоваться, не использовали «родной» интерфейс Win32. Не знаю, чем руководствовались разработчики, но с точки зрения конечного пользователя это выглядит очень не симпатично.  
  
Примеры приложений: Eclipse, ZDE, клиент для Gnutella Limewire.  
Плюсы: мультиплатформенность, большое количество кадров, развитость фреймворка.  
Минусы: необходимость установки фреймворка, кривость GUI, низкая производительность.   
C#   
Язык C# и платформа .NET вышли из под крыла компании Microsoft, которая никогда не отличалась уважением к мультиплатформенности. Данный язык был тепло принят разработчиками и энтузиасты проекта Mono даже сделали аналог фреймворка под Linux/Unix/Mac Os X.  
  
Аналогично с Java, приложения на C# имеют большой минус — необходимость устанавливать фреймворк. Я сам отказался от установки несколько приложений, которые требовали этого фреймворка. Менее продвинутые пользователи будут ещё более требовательными.  
  
В остальном — сплошные плюсы, на мой взгляд.  
  
Примеры приложений:?  
Плюсы: мультиплатформенность, большое количество кадров, хорошая производительность, развитость фреймворка.  
Минусы: необходимость установки фреймворка.   
C++   
Старичок дотянул до наших дней и замечательно себя чувствует. Много приложений под платформы Linux и Windows до сих пор пишутся на этом языке.  
  
Программы, написанные на C++,

являются примером для других по размеру дистрибутива и экономному использованию системных ресурсов (процессор, память). Тем не менее у разработчиков есть масса претензий к С++. По моему мнению, язык является «устаревшим» и его популярность в дальнейшем будет снижаться, что подтверждается индексом TIOBE.  
  
С точки зрения развития проекта, по сравнению с динамическими интерпретируемыми языками (вроде Ruby и Python), разработка на данном языке может иметь менее высокую скорость и более высокие издержки изменения проекта. Для стартапа, которому не столь важна производительность приложения, это может стать существенным минусом.  
  
Примеры: Firefox,?..  
Плюсы: отличная производительность, большое количество кадров, большое количество библиотек.  
Минусы: невысокая скорость разработки.   
Python   
Взглянуть на Python в качестве платформы для desktop-приложений меня заставила программа MusicBrainz Picard. Несмотря на свою скриптовую сущность, Python легко собирается в один exe-файл, не требуя от пользователя установки дополнительных компонентов.  
  
В случае разработки небольшого приложения, интерпретируемые языки вроде Python будут большим плюсом. Легкость написания и высокая скорость изменения приложения пригодится любому стартапу.  
  
Огромным минусом различных «модных» технологий является их низкое распространение, а значит серьезные проблемы в поиске квалифицированных кадров. Ситуация с поиском программистов итак плачевная, а если ограничиться узким языком — можно вообще никого не найти. С другой стороны, храбрость перейти на новый язык имеют наиболее прогрессивные разработчики. Может получиться так, что выбрав «перспективный» язык, мы сразу отсечем миллионы середнячков, оставив себе выбор из нескольких перспективных разработчиков.  
  
Пример: MusicBrainz Picard, оригинальный BitTorrent.  
Плюсы: высокая скорость разработки и изменений, хорошая интеграция с библиотеками на С и С++.  
Минусы: мало кадров, низкая производительность.   
Выводы   
К сожалению, любая из вышеперечисленных платформ имеет свои плюсы и минусы, однозначного решения найти не удалось. Выбор одной из них сегодня принесет преимущества и недостатки, влияние которых на проект мы увидим только завтра.

**День 3**

Введение

Стремительное развитие компьютерной техники в последние годы, появление мощнейших графических ускорителей и центральных процессоров способствовало не менее бурному развитию индустрии компьютерных игр. Выдающиеся разработки этой отрасли – это сложнейшие программы, как правило, с очень высокими требованиями к аппаратной части компьютера. Однако для возможности отдохнуть в перерыве от выполнения какой-либо работы оператору компьютера не всегда требуется новейшая компьютерная игра, а зачастую использовать её не позволяет маломощное оборудование офисного компьютера

При всей кажущейся простоте, популярная игра «крестики-нолики» требует повышенного внимания и отлично способствует развитию логики. В эту игру играют не только дети, есть любители поиграть в «крестики-нолики» и среди взрослых; дети учатся в этом случае основам логического мышления, а взрослые вспоминают его базу и получают разрядку на фоне напряжённых трудовых будней.   
  
Играть в «крестики-нолики» можно с товарищем или с виртуальным партнёром, выбрав для этого желаемый уровень сложности. Такое времяпровождение невозможно назвать бесцельным, ведь оно помогает расслабиться с одной стороны и настраивает на позитивное мышление с другой.

Основания для разработки;

Основанием для разработки программного продукта «Хорошко и крестики-нолики» является задание руководителя практики Хорошко М.Б., в соответствии с планом учебной практики и заданий которые он нам скидывает в вк.

Игра рассчитана на одного/двух игроков.

Назначение разработки;

Получить опыт в создании приложения на мобильное устройство в связи с прохождением практики в ВУЗе.

Данный проект является игрой, вследствие чего предусматривается одна категория пользователей - игроки. В процессе работы приложения пользователь является непосредственным участник игрового процесса и оказывает непосредственное влияние на него.

Программы должна обладать следующим функционалом:

а) звуковой функционал:

1) отключение, включение музыки;

б) внутриигровой функционал:

1) выбор размера игрового поля;

2) подтверждение действия;

в) интерфейс пользователя:

1) игровые поля разных размеров;

2) главное меню;

требования к программе или программному изделию;

1.Требования к функциональным характеристикам:

–Приложения Android ;

–Совместимость с Android : Android 4.4. и старше ;

–Верстка телефон книжная : Да ;

–Верстка телефон альбомная : Да ;

–Верстка планшет Книжная : Адаптивная от телефона ;

–Верстка планшет Альбомная : Адаптивная от телефона · требования к надежности;

–Объем приложения до 1Гигабайта;

–Совместная игра 2х игроков;

–Программа должна иметь простой графический интерфейс.

2.Условия надежность:

Перечень требований, для обеспечения надежного функционирования программы:

Необходимый уровень квалификации разработчиков - программист обязан знать принципы ООП, уметь работать с классами и объектами, реализовывать алгоритмы на языке Java, взаимодействовать с интерфейсом разрабатываемого приложения;

регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие вирусов;

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через Веб интерфейс недопустимы;

организацией бесперебойного питания технических средств;

Время на восстановление приложения после отказа работы приложения, не должно превышать времени повторного запуска приложения. Время на восстановление приложения после отказа устройства, не должно превышать времени на перезагрузку OS Android и запуска приложения.

3. Условия эксплуатации:

Условия эксплуатации соответствуют условиями эксплуатации мобильного устройства.

4.Требования к составу и параметрам технических средств;

–Смартфоны;

–Операционная система Android OS 4.0;

–API 17 и выше;

– Любое разрешение экрана;

– Возможность получения файла приложения (.apk) через интернет, bluetooth или usb-кабель.

5. Требования к информационной и программной совместимости;

отсутствуют

6. Требования к маркировке и упаковке;

отсутствуют

7. Требования к транспортированию и хранению;

Не нагревать

8. Специальные требования.

отсутствуют

требования к программной документации;

Состав программной документации должен включать в себя:

1. техническое задание;

2. программу и методики испытаний;

3. руководство пользователя;

технико-экономические показатели;

Ориентировочная экономическая эффективность не рассчитываются. Аналогия не проводится ввиду уникальности предъявляемых требований к разработке.

стадии и этапы разработки;

Разработка должна быть проведена в три стадии:

1. Стадия «Технический проект»

* Разработка технического проекта
* Утверждение технического проекта

1. рабочее проектирование;

* Выбор технологий и способа реализации
* Создание серверной части
* Создание Android-интерфейса
* Тестирование продукта

1. внедрение.

* разработка плана мероприятий по внедрению

порядок контроля и приемки;

Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:

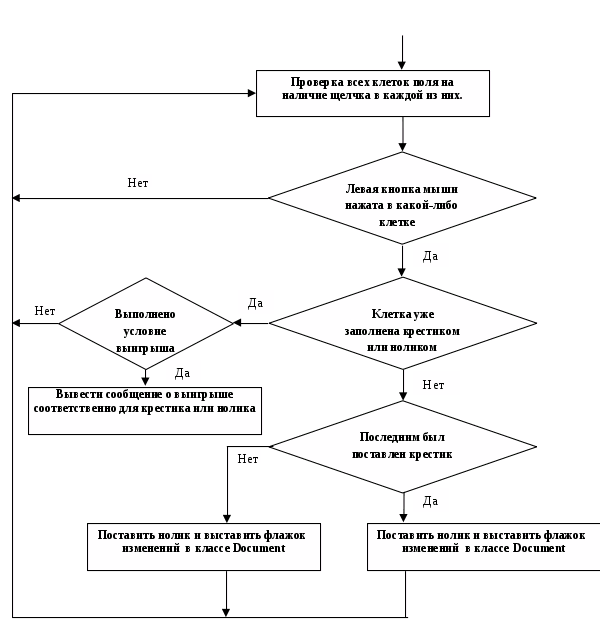
приобрести компоненты технического и программного обеспечения, заключить договора на их лицензионное использование;

завершить работы по установке технических средств;

провести обучение пользователей.

в техническое задание допускается включать приложения.

Блок-схема алгоритма:



# День 4

# ПРОТОТИПИРОВАНИЕ И ЕГО РОЛЬ В РАЗРАБОТКЕ

Правильное размещение элементов интерфейса повышает его юзабилити, делает ресурс более привлекательным для пользователей. В отдельных случаях, с помощью грамотной компоновки сайта можно склонить посетителей к определенным действиям: оставить заявку, подписаться на рассылку, приобрести товар и т.п. Нужно четко представлять, чего вы ожидаете от ресурса, какие элементы должны присутствовать на каждой странице.

С этой целью на этапе проектирования предварительно создают прототип — черно-белый макет, представляющий упрощенную схему сайта. Он содержит все основные элементы, представленные в виде блоков, поэтому клиент может оценить основную концепцию.

Прототипирование помогает не только выявить и сформулировать основное направление дизайна, но и значительно сэкономить время. Потратив день на создание концепта, вам не придется тратить неделю на разработку нового макета сайта. Но это не единственная причина, по которой стоит уделить время прототипу.

## Причины, по которым стоит делать прототипы

Прежде чем перечислить преимущества прототипирования, нужно четко определить, что собой представляет данный процесс.

Прототип — важный эскизный проект, который предшествует разработке оригинального дизайнерского макета. Он призван не только показать структуру будущего сайта, размещение основных элементов интерфейса, но и карту сайта, взаимосвязь его основных страниц.

Прототипирование является процессом, призванным значительно сократить время [разработки сайта](https://depix.ru/), благодаря фокусировке проектировщика на основных функциональных и маркетинговых факторах.

### Среди главных причин использования прототипирования перед созданием макета можно выделить следующие:

* прототип дает заказчику полноценное представление о том, как будет выглядеть сайт в конечном результате;
* прототипирование позволяет рационализировать процесс разработки дизайна, концентрирую внимание на важных элементах интерфейса;
* эскиз сайта – незаменимая вещь, если клиент еще не до конца понимает, что именно он ожидает от сайта, какие функции будет нести каждая страница. Тщательное планирование на этапе прототипирования дает возможность избежать глобальных изменений в готовом макете;
* на этапе прототипирования можно выявить ненужные элементы, от которых лучше всего отказаться, или же наоборот — дополнить интерфейс дополнительными деталями;
* данный процесс значительно снижает объем работы дизайнера по разработке проекта, а значит и экономит деньги заказчика;
* имея на руках прототип, дизайнер и заказчик более четко представляют конечный результат;
* разработка прототипа предполагает вовлечение заказчика, способствует более продуктивной работе, согласованности процесса.

При этом прототип создается очень просто. Отдельные виды можно набросать уже при первой встрече с заказчиком, уточняя определенные детали проектирования.

## Основные виды прототипов

Создавать эскиз сайта можно разными способами. Одним из самых простых и быстрых прототипов — нарисованный от руки. Несмотря на развитие технологий, бумажные модели пользуются большой популярностью среди дизайнеров. Преимуществом данного вида является скорость. Создать набросок модели можно уже на этапе обсуждения, внося поправки заказчика. Это дает возможность лучше понять клиента, определить цели проекта. Чаще всего бумажные прототипы используются на этапе проработки идеи и возможных вариантов, ведь нарисовать несколько возможных прототипов гораздо проще, чем создавать их с помощью приложений.

Бумажное прототипирование относится к статичным моделям, которые отличаются содержанием статичных изображений. Помимо концептов, нарисованных от руки, к статичным прототипам относятся эскизы, созданные в графических редакторах, нарисованные на планшете или маркерной доске. Они отображают исключительно проект дизайна сайта.

В отличие от статических, интерактивные прототипы представляют собой проектирование взаимодействия всех составляющих ресурса. Это упрощенные макеты всех страниц вэбсайта с высокой детализацией. При этом все элементы кликабельны. Вы можете перейти со страницы на страницу, развернуть меню и т.п. Интерактивный прототип помогает лучше понять механизм работы проекта, осуществить поставленные проектные решения, выявить целесообразность их реализации в готовом макете.

## Инструменты для создания прототипа

Современные инструменты позволяют создавать интерактивные макеты легко и достаточно быстро. Для этого не требуется глубоких знаний верстки.

Каждый инструмент обладает своими особыми возможностями и преимуществами. Выбрать подходящее приложение вы сможете, оценив поставленные цели и задачи проектирования. Для выбора того или иного инструмента, оцените простоту его использования, совместимость с вашей операционной системой, возможность отправить прототип другому участнику группы для анализа, внести правки или полностью изменить исходный вариант. Поскольку инструмент призван упростить процесс моделирования, программа должна содержать достаточное количество шаблонов и трафаретов, давать возможность адаптировать макет под разные устройства.

### Ниже приведены популярные инструменты для создания прототипа:

1. Omnigraffle (приложение для OSX) — создает графические схемы, соединяя отдельные элементы линиями. Позволяет выстраивать диаграммы, содержит элементы моделирования, экспорта и импорта готовых файлов. Приложение легко освоить даже начинающему дизайнеру

2. ConceptDrawPro (приложение для Windows) — графическое приложение, которое используется для создания бизнес-проектов, прототипов и диаграмм, проектных документов.

3. Pidoco (онлайн приложение) — программное обеспечение, позволяющее создавать прототипы сайтов в один клик. Простой интерфейс делает роботу с Pidoco простой и понятной, а результат можно протестировать на любом мобильном устройстве.

4. Mockingbird (онлайн приложение) — помогает создавать интерактивные модели будущего сайта или программного обеспечения в режиме реального времени. Вы можете легко связать страницы между собой, посмотреть предварительный результат, внести правки и корректировки.

5. ProtoShare (онлайн приложение) — удобный инструмент, дающий возможность создавать динамические изображения, интерактивные прототипы. Преимуществом приложения является возможность работать над проектом группой, просматривать результаты и оставлять комментарии.

## Упрощение процесса разработки при использовании прототипов

Сам по себе процесс создания прототипа достаточно простой. Для упрощения его условно делят на три составляющие:

1. Создание — простейший эскиз, основываясь на пожеланиях заказчика. Используются при этом основные принципы построения интерфейса и юзабилити;

2. Оценка — обсуждение с клиентом насколько точно мы поняли его требования и пожелания;

3. Доработка — доработка проблемных участков эскиза или изменение макета полностью.

Первичный прототип может состоять только из общих блоков с минимальной детализацией. В процессе разработки, он будет «обрастать» дополнительными элементами, постепенно приближаясь к итоговому макету.

Играет важную роль скорость данного процесса — время между итерациями нужно минимизировать. В целом на создание и доработку прототипа может уйти от пары минут (например, для наброски в процессе обсуждения заказа с клиентом) до нескольких дней. В любом случае, время, потраченное на прототипирование, поможет выполнить работу быстро и более качественно.

**Вывод**

Главная цель прототипирования — передать аудитории суть вашей идеи с минимальными затратами сил и времени. А значит, вы страхуете себя и свои деньги от неудачи.

Книгу определенно стоит прочитать людям, которые впервые узнали о процессе прототипирования. Иначе, вам достаточно было прочитать эту заметку. Ничего нового, кроме примеров из жизни автора не узнаете.

**День 5**

Чтобы начать работать с [Python 3](https://python-scripts.com/), вам нужно получить доступ к интерпретатору Python. Существует несколько общих способов сделать это:

* Python можно получить на сайте Python Software Foundation [python.org](https://www.python.org/). Как правило, это означает загрузку нужно установочного файла для вашей операционной системы и запуска его на вашем компьютере.
* Некоторые операционные системы, особенно Linux, предоставляют менеджер пакетов, который можно запустить для установки Python.
* Для macOS, лучший способ установить Python 3 включает в себя установку менеджера пакетов под названием Homebrew. Вы увидите, как это делать в соответствующем разделе статьи.
* Для мобильных операционных систем, таких как Android и iOS, вы можете установить приложения, которые предоставляют среду программирования Python. Это может быть отличным способом практики ваших навыков на ходу.

В качестве альтернативы, существует ряд сайтов, которые позволяют вам получить доступ к интерпретатору Python онлайн, без необходимости в установки чего-либо на вашем компьютере. Прежде чем начать изучать [основы](http://www.internet-technologies.ru/articles/obuchenie-programmirovaniyu-s-nulya-s-chego-nachat.html) языка программирования Python вы должны иметь стабильную версию интерпретатора Python.

## Установка Python 3 на Windows

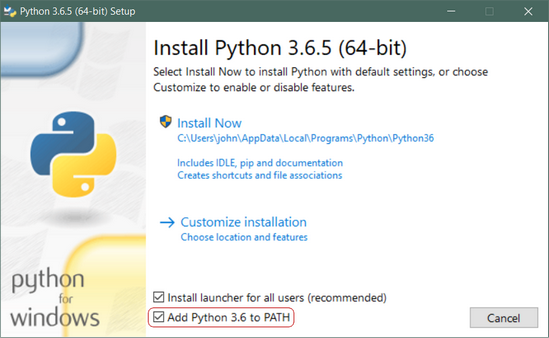
Вероятность того, что на вашей системе Windows заранее установлен Python – крайне мала. К счастью, установка не требует большего, чем загрузка установочного файла Python с сайта python.org и его запуска. Давайте рассмотрим, как устанавливать Python 3 на Windows:

### Шаг1 : Загрузка установочного файла Python 3

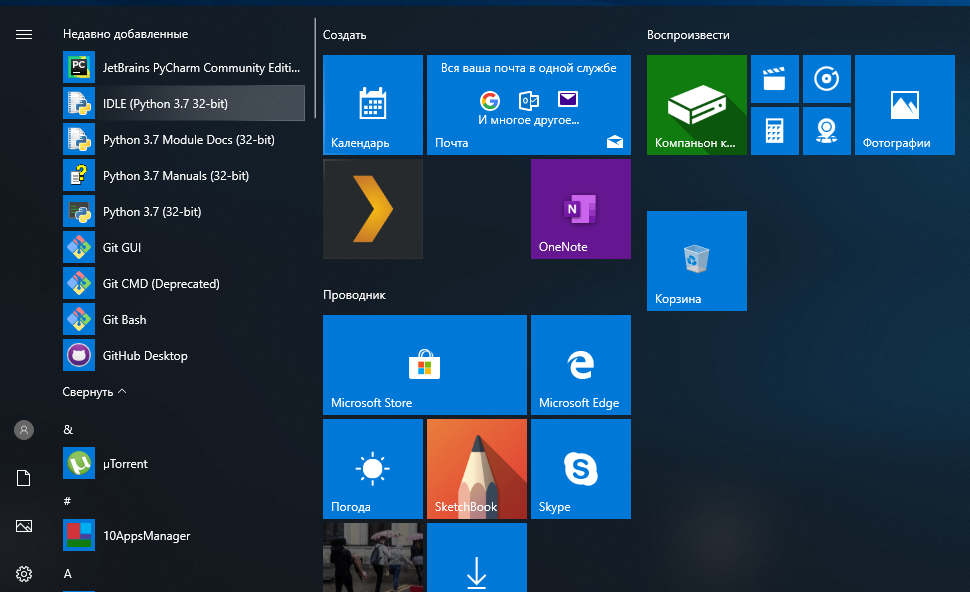
1. Открываем окно браузера и переходим на страницу [Download](https://www.python.org/downloads/windows/) для Windows на python.org;
2. Под верхним заголовком, где написано Python Releases for Windows, нажимаем на ссылку к последней версии Python 3.x.x.;
3. Листаем вниз и выбираем установочный файл Windows x86-64 для 64-разрядной версии операционной системы или файл Windows x86 для 32-разрядной (Смотреть ниже).

## Шаг 2: Запускаем установочный файл

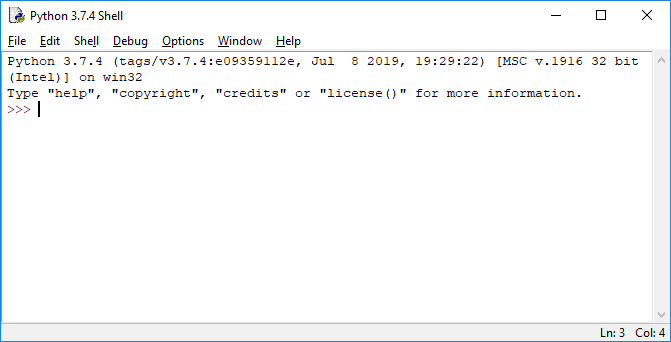
После того, как вы выбрали установочный файл и загрузили его, просто запустите его двойным нажатием на загруженный файл. Должен открыться диалог, который выглядит следующим образом:

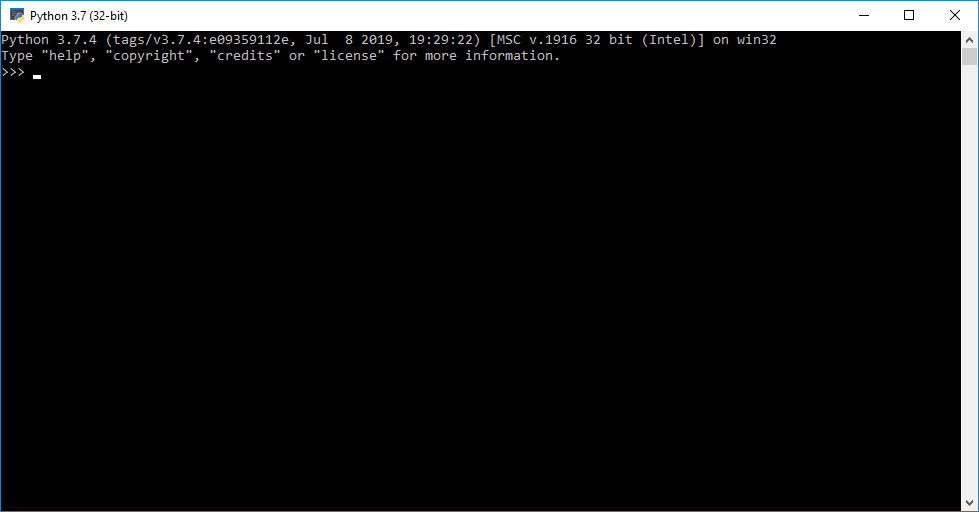
Далее, просто нажмите на Install Now. Это все, что нужно.

Установленные программы:



Запуск программы:





**День 6**

**Внешний вид и selector’ы**

Стандартные контролы вызывают у пользователей скуку, поэтому, практически все приложения в android market’e, обладают своими дизайнерскими решениями. Корпорация Google создала для этого мощный описательный инструмент стилей и элементов, основанный на xml. С его помощью можно задавать общий стиль групп элементов и рисовать свои новые контролы.

Для начала создадим в среде Eclipse новое приложение и назовем его Tic Tac Toe, установим в качестве **Build Target** — Android 1.6 и **Create activity**: TicTacToe.

Автоматически будет создан файл TicTacToe.java. Это activity, она будет вызываться при запуске приложения. Среда Android узнает об этом из файла AndroidManifest.xml, где для нашей activity задан <intent-filter>:

<activity android:name=".activities.TicTacToe"

android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

Перейдем непосредственно к созданию интерфейса. В качестве корневого элемента main.xml мы положим ScrollView, этот элемент позволит нам прокручивать нашу activity, если контролы в горизонтальном положении телефона выйдут за пределы формы. Следующим элементом идет LinearLayout, благодаря ему мы разместим наши кнопки стройно в ряд, задав все необходимые отступы.

Теперь сделаем <selector>, для того чтобы задавать правила отображения наших кнопок. Этот xml файл будет содержать данные о том, как рисовать контрол в нажатом, сфокусированном или обычном положении.

button\_selector.xml

<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http:**//**schemas.android.com/apk/res/android">

<item android:state\_pressed="true">

<shape>

<solid

android:color="#774401" />

<stroke

android:width="1dp"

android:color="#171717" />

<corners

android:radius="4dp" />

<padding

android:left="10dip"

android:top="10dip"

android:right="10dip"

android:bottom="10dip" />

</shape>

</item>

<item>

<shape android:layout\_marginBottom="25dip">

<gradient

android:startColor="#9E5A02"

android:endColor="#2A1800"

android:angle="270" />

<stroke

android:width="1dp"

android:color="#171717" />

<corners

android:radius="6dp" />

<padding

android:left="10dip"

android:top="10dip"

android:right="10dip"

android:bottom="10dip" />

</shape>

</item>

</selector>

Наш selector содержит описание нажатого состояния и состояния покоя. Во втором случае мы заливаем кнопку градиентом, задавая startColor и endColor. Сorners указывает на углы нашей кнопки. Для эффекта нажатого состояния мы уменьшаем угол скругления и заменяем градиент монотонным цветом. Все имена атрибутов легко читаемы, их полный список можно посмотреть на официальном сайте [Shape Drawable](http://developer.android.com/guide/topics/resources/drawable-resource.html#Shape).

Чтобы не описывать, каким будет размер шрифта и стиль текста в каждой кнопке, мы зададим один общий для всех, создав в папке values файл styles.xml.

<?xml version="1.0" encoding="utf-8"?>

<resources>

<style name="ButtonStyle">

<item name="android:layout\_width">fill\_parent</item>

<item name="android:layout\_height">wrap\_content</item>

<item name="android:textColor">#ffffff</item>

<item name="android:gravity">center</item>

<item name="android:layout\_margin">4dip</item>

<item name="android:minHeight">48dip</item>

<item name="android:textSize">18dip</item>

<item name="android:textStyle">bold</item>

<item name="android:shadowColor">#000000</item>

<item name="android:shadowDx">1</item>

<item name="android:shadowDy">1</item>

<item name="android:shadowRadius">2</item>

</style>

</resources>

Тогда xml текст кнопки будет простой:

<Button

android:id="@+id/new\_game\_button"

android:text="@string/new\_game\_label"

android:layout\_width="fill\_parent"

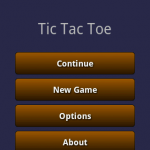
android:layout\_height="wrap\_content"

android:background="@drawable/button\_selector"

style="@style/ButtonStyle"

/>

В итоге у нас получается интересная первая activity:

[](http://www.4htc.ru/wp-content/uploads/2011/09/0_5994b_b9e677b3_L.png)

**Стратегия игры**

Классические «крестики-нолики» на поле 3×3 имеют достаточно короткое описание всех вариантов состояний. При совершенной игре результат всегда будет сведен к ничьей, если один из соперников не совершит ошибку. Для игры за компьютер (нолики) мы будем использовать простой принцип приоритета клеток. Максимальным приоритетом будет обладать центральное поле. Вторыми по очереди будут поля в углах, и самые *нежеланные*будут оставшиеся поля в середине.

Весь алгоритм можно описать так:

* Если можно сделать шаг ведущий к выигрышу — делаем. Иначе,
* Если нужно блокировать возможность выигрыша соперника — блокируем. Иначе
* Делаем ход в ячейку с максимальным приоритетом.

Три уровня сложности будут такими, для **Hard**придерживаемся вышеописанного алгоритма, к слову, он упрощен и обыграть его имеются возможности. Для **Easy**будем ставить нолик в случайную ячейку, а для **Medium**мы будем делать в такой пропорции: один из трех будет глупый ход, два из трех — умные.

Tic-tac-toe layout

Создадим game.xml layout для отображения процесса игры. Мы будем использовать TableLayout с тремя строками и столбцами для поля 3×3. В каждой ячейке будет лежать кнопка с пустой или заполненной картинкой.

<TableLayout android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" xmlns:android="http:**//**schemas.android.com/apk/res/android"

android:id="@+id/buttons" android:background="@color/background">

<TableRow

android:id="@+id/row1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="1dip"

android:layout\_marginRight="1dip"

android:layout\_marginBottom="2dip">

<Button

android:text=""

android:id="@+id/button\_11"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/clear"

android:layout\_marginRight="2dip"

android:layout\_marginLeft="12dip">

...

<Button

android:text=""

android:id="@+id/button\_33"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/clear">

</Button>

</TableRow>

…

TableRow обеспечивает горизонтальное распределение всех принадлежащих ему контролов. Чтобы наша сетка не сливалось в единое целое,  зададим необходимые отступы для каждой ячейке layout\_margin Right, Left, Top или Bottom. При нажатии на пустую кнопку, мы будет рисовать в ней «крестик». Ход компьютера будет рисовать ответный «нолик».

Model-view-controller

Теперь немного об архитектуре нашей игры. Для реализации режима битвы и возможности продолжить играть после закрытия activity, нам нужно будет хранить текущее состояние (счет, заполненные ячейки). В этом нам помогут такие шаблоны проектирования как model-view-controller для разделения логики и singleton для хранения состояния. Шаблон MVC позволит разделить данные, представление и обработку действий пользователя на три отдельных компонента, что облегчит реализацию алгоритма игры и дальнейшее сопровождение программы.

В модели (см. TicTacToeModel.java) мы реализуем два метода, doSmartMove() для хорошего хода и doStupidMove() для случайного. В случае, когда уровень сложности выбран Medium, мы будем делать два умных и один глупый ходы. Модель также будет хранить счет выигранных и проигранных боев.

Задача controller’а обновлять ход игры по запросу, а в роли view выступает сама Activity.

**Локализация**

География Android устройств обширна, поэтому вопрос локализации занимает ключевое место при разработке приложений под эту платформу. К счастью, нам не прийдется изобретать велосипед, Google позаботился о разработчиках в этом вопросе.

Правила локализации приложений таковы — мы создаем несколько наборов ресурсов, первый — по умолчанию res/values/strings.xml, остальные для нужной нам локали res/values-<qualifiers>/strings.xml, например, values-en для английского или values-ja для японского. При запуске activity Android выбирает какие ресурсы загрузить, основываясь на локали самого устройства. К слову, под ресурсами в Android подразумеваются не только текстовые строки, а также layout’ы, звуковые файлы, графика и другие статические данные.

Для нашего приложения мы создадим папки res/values, res/values-en и res/values-ru, в них будут лежать файлы strings.xml. Эти файлы содержат название приложения, сообщения и вопросы. Английский файл выглядит так:

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="app\_name">Tic Tac Toe</string>

<string name="main\_title">Tic Tac Toe</string>

<string name="continue\_label">Continue</string>

<string name="new\_game\_label">New Game</string>

<string name="options\_label">Options</string>

<string name="easy\_label">Easy</string>

<string name="medium\_label">Medium</string>

<string name="hard\_label">Hard</string>

<string name="message\_title">Message</string>

<string name="question\_title">Question</string>

<string name="draw\_game">Draw!</string>

<string name="nought\_win\_game">Nought win!</string>

<string name="cross\_win\_game">Cross win!</string>

<string name="restart\_game">Restart Game?</string>

<string name="about\_label">About</string>

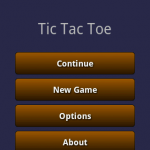
<string name="about\_title">About Tic Tac Toe</string>

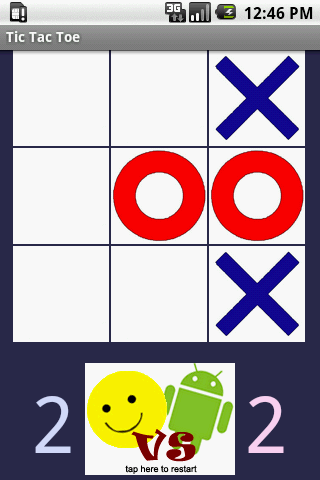
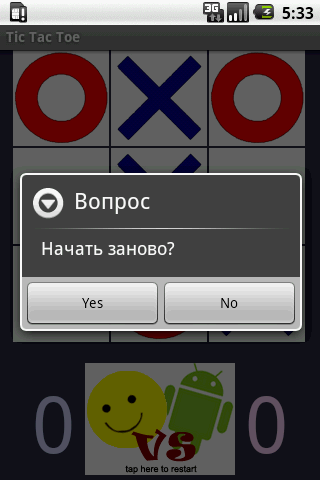
...

</resources>

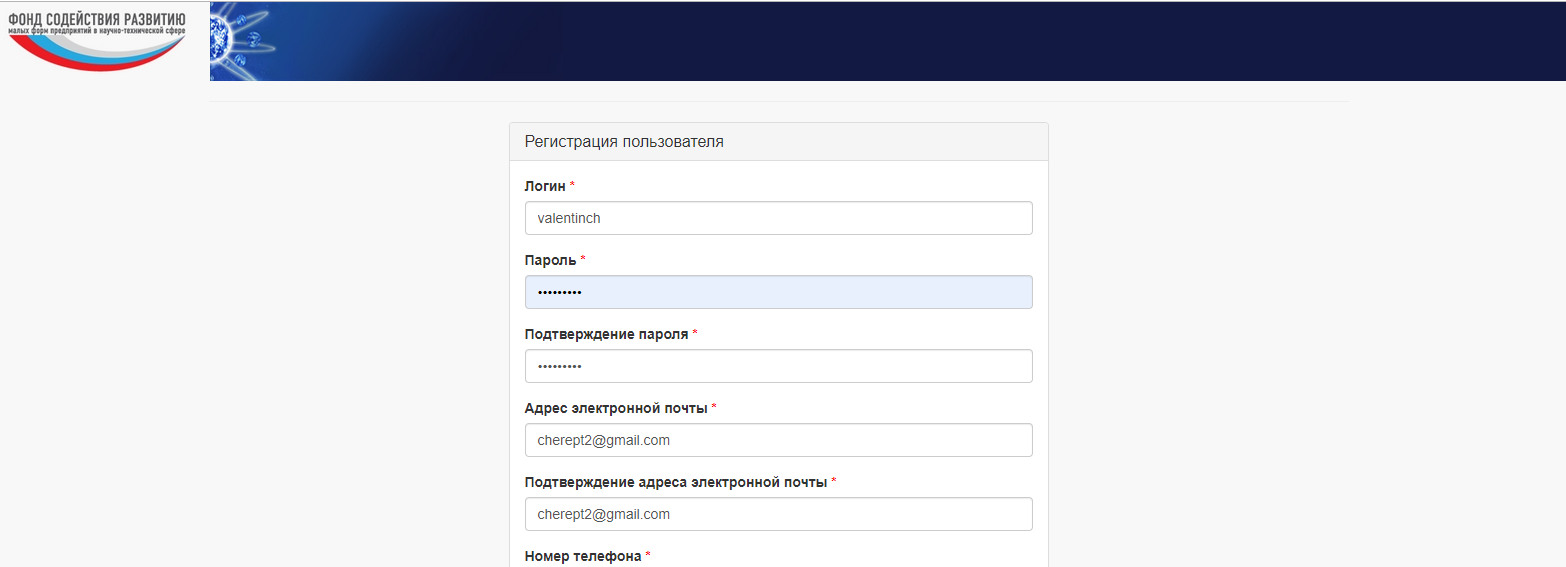
Еще нюанс, android developer’s guide предупреждает нас о необходимости иметь точную копию какой-нибудь локализации в качестве ресурса по умолчанию, чаще всего — английскую. И приводит пример, что если не будет хватать строки в файле res/values/strings.xml, которая есть в res/values-en/strings.xml и используется в приложении, то возможно все будет компилироваться без проблем, но в локали отличной от английской пользователь увидит сообщение об ошибке и кнопку о закрытии приложения.

В итоге у нас получились вот такие локализованные крестики-нолики:

[](http://www.4htc.ru/wp-content/uploads/2011/09/0_5994b_b9e677b3_L.png)

[](http://www.4htc.ru/wp-content/uploads/2011/09/0_5994c_2f000f8a_L.png)[](http://www.4htc.ru/wp-content/uploads/2011/09/0_5994d_e4e63bbf_L.png)[](http://www.4htc.ru/wp-content/uploads/2011/09/0_59a03_7b395661_L.png)

**День 7**

**Регистрация:**

