Программирование мобильных приложений в MIT App Inventor

Практикум

Авторы:

Ливенец Марина Александровна
Ярмахов Борис Борисович

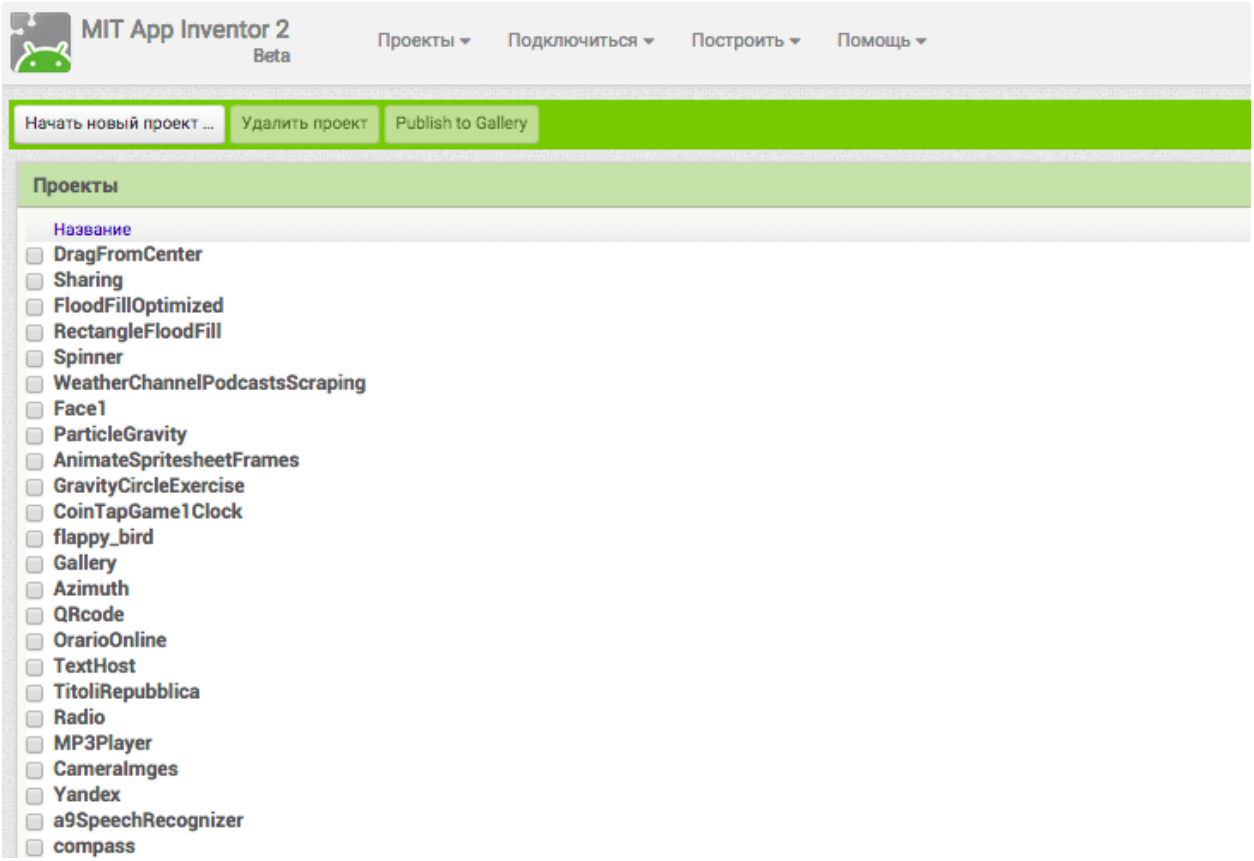
Оглавление

Глава 1. Введение. Среда MIT App inventor. Интерфейс пользователя

MIT App Inventor (http://ai2.appinventor.mit.edu/) - облачная среда визуальной разработки приложений для платформы OS Android, работа в которой не требует знания языка программирования Java и Android SDK, достаточно знания элементарных основ алгоритмизации. Для работы в MIT App Inventor необходимо наличие Google или Google Apps аккаунта, а построение программ осуществляется в визуальном режиме с использованием блоков программного кода.

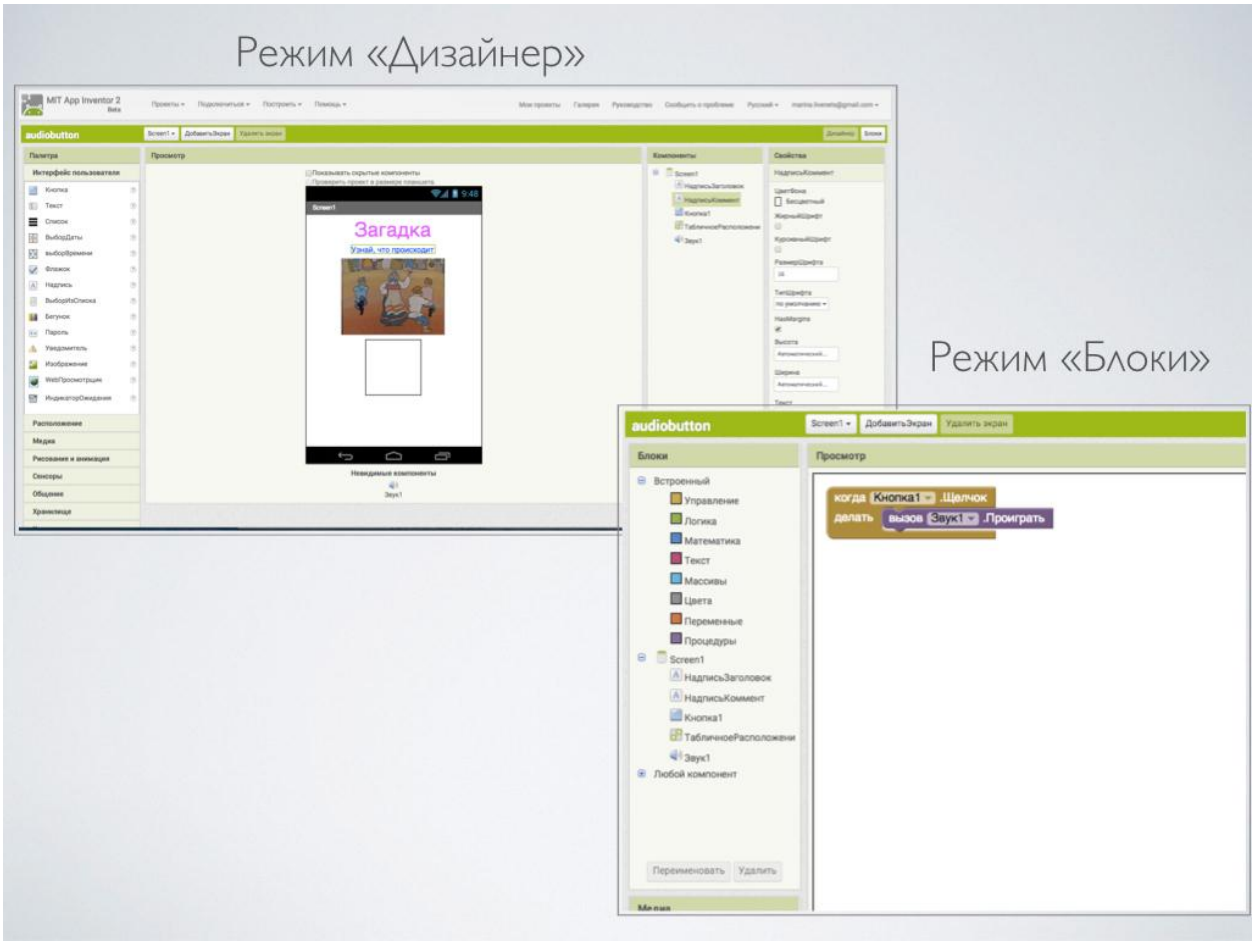
1.1 Описание интерфейса пользователя

После входа в MIT App Inventor пользователь попадает на страницу, где отображается список его проектов. Все созданные проекты хранятся в папке Мои проекты.



Вновь создаваемое приложение в среде MIT App Inventor это новый проект (меню Проекты - Начать новый проект). Нажатие кнопки “Удалить проект” - удаляет созданный проект. Переход к списку проектов из окна разработки по по ссылке “Проекты/Мои проекты».

Разработка мобильного приложения в MIT App Inventor http://ai2.appinventor.mit.edu/ происходит в 2 этапа. Первый этап - проектирование интерфейса пользователя “Как это будет выглядеть”, второй - программирование компонент приложения “Как они будут себя вести”.



Эти два процесса реализуются в отдельных окнах, по сути это два разных режима работы в среде с MIT App Inventor.

1.2 Режим “Дизайнер”

Режим “Дизайнер” - режим в котором создается интерфейс (“внешний вид”) приложения” . Данный режим используют для выбора и размещения различных компонент приложения: кнопок, текстовых полей, изображений и др., которые отображаются на экране вашего устройства, при запуске приложения.

Интерфейс для разработки дизайна проекта состоит из следующих основных элементов:

- **Палитра** включает наборы (группы) компонент будущего приложения.

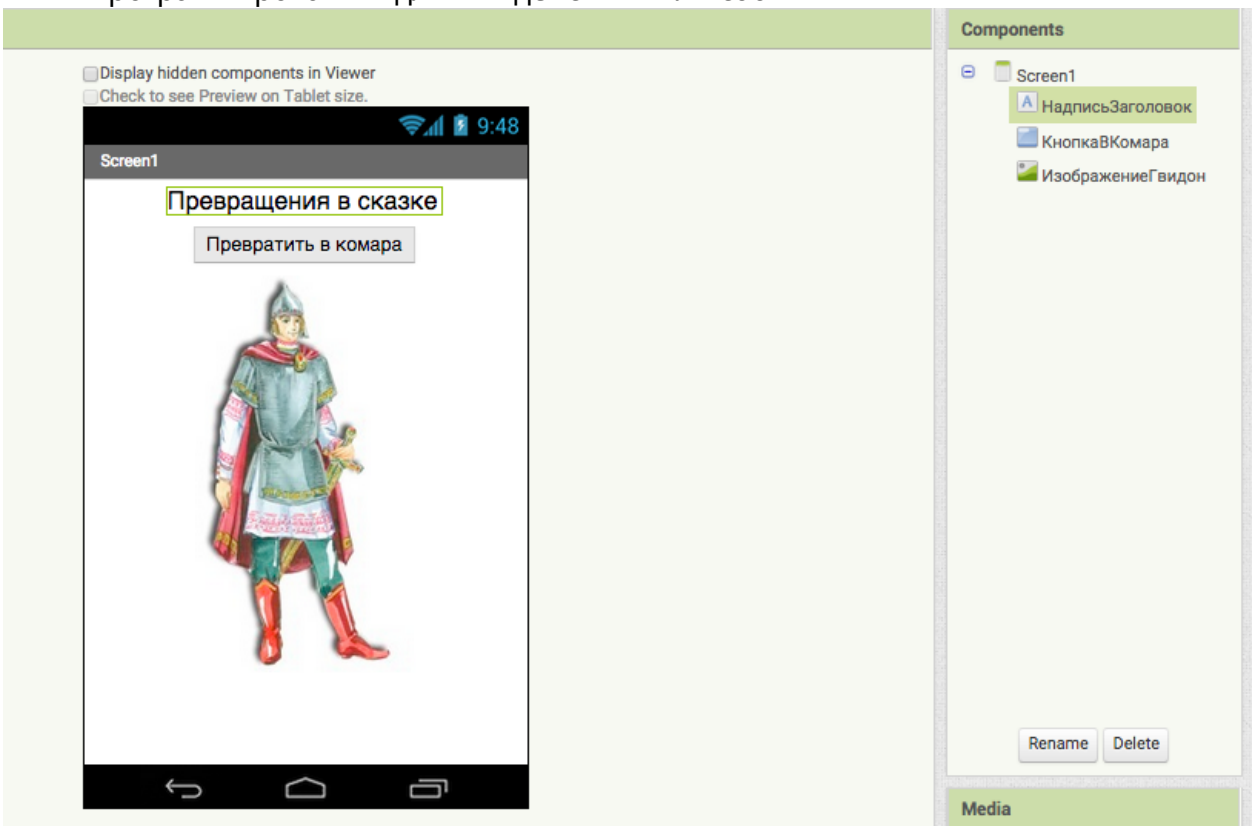
Компонентами называются функциональные элементы приложения, такие как кнопки, изображения, текст, поля для ввода текста, дат, интерфейсы для подключения к разным датчикам вашего Android-устройства — акселерометр, GPS, базы данных и др. Некоторые компоненты являются частью графического дизайна, например, кнопки, а некоторые — невидимы на экране устройства, например, таймер, сенсоры или видеоплейер.

Просмотр — экран вашего приложения. Точнее один из экранов. В приложении можно использовать несколько экранов, где будут производиться различные действия. Например, на первом экране у вас инструкция к приложению а на втором экране, собственно, приложение — его функциональная часть.

Компоненты — здесь расположен список компонентов, которые вы уже используете в своем проекте.

Именование компонентов приложения

При именовании компонентов рекомендуется воспользоваться следующим правилом ***"Имя компонента" = " Название компонента"+ "Действие/ Функция***", которое он выполняет в приложении: КнопкаНазад, КнопкаДалее, ИзображениеФон и пр. Такое именование компонент позволяет легко ориентироваться при программировании для них действий или событий



Свойства — в этой части экрана устанавливаются свойства компонент вашего приложения, например: цвет, размер шрифта, источники изображений и звуков, надписи, первоначальное значение и другие.
Медиа — список используемых медиафайлов (изображений, видео-, аудио-роликов и т.п.)

О размерах используемых файлов

Мобильное приложение - это не тот случай, когда в него внедрены фильмы или большие коллекции аудио записей. Желание встроить в приложения большие графические и звуковые файлы, может привести к тому, что оно не только будет долго загружаться, но и займет достаточно много памяти на самом мобильном устройстве.

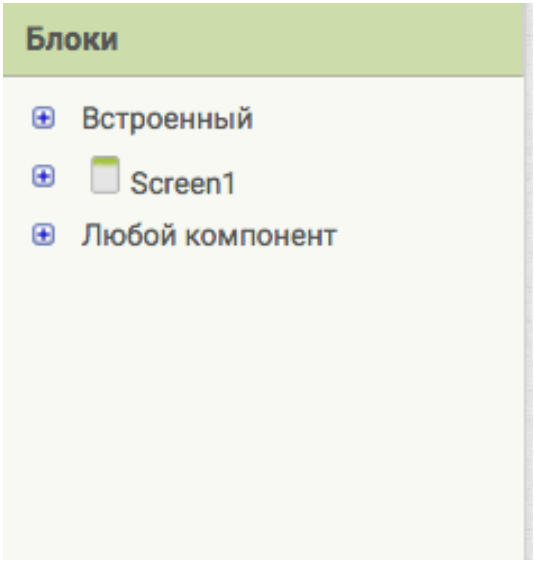
1.3 Экраны приложения

Проект приложение может состоять из множества экранов. Для работы с экранам «Screen», в окне разработки есть кнопки добавления экрана — «Добавить Экран и удаления экрана — “Удалить Экран”. Запуск приложения всегда начинается со стартового экрана, дизайн которого может включать набор компонент для перехода на другие экраны.

В среде MIT App Inventor количество экранов не должно превышать 10. При создании 11 экрана выдается предупреждение о превышении допустимого количества экранов.

1.4 Режим “Блоки”

Режим “Блоки” используется для программирования поведения вашего приложения и его компонент, каким образом выбранные вами компоненты, будут реагировать на различные действия пользователя.

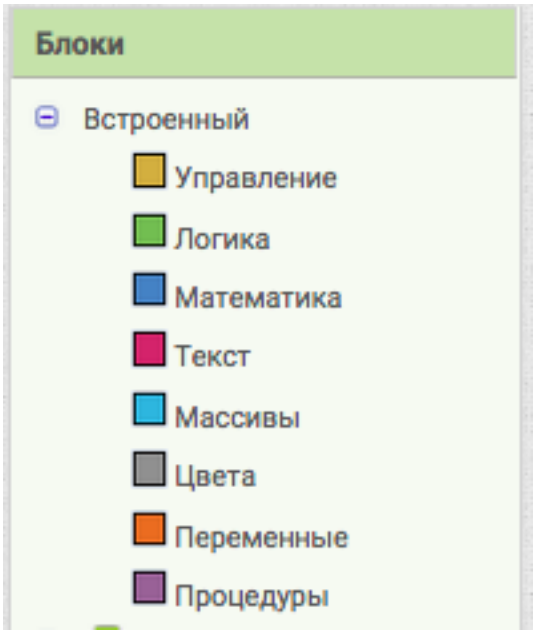


В режиме “Блоки” используются три группы Блоков:

Основные группы Блоков используемых при создании приложений:

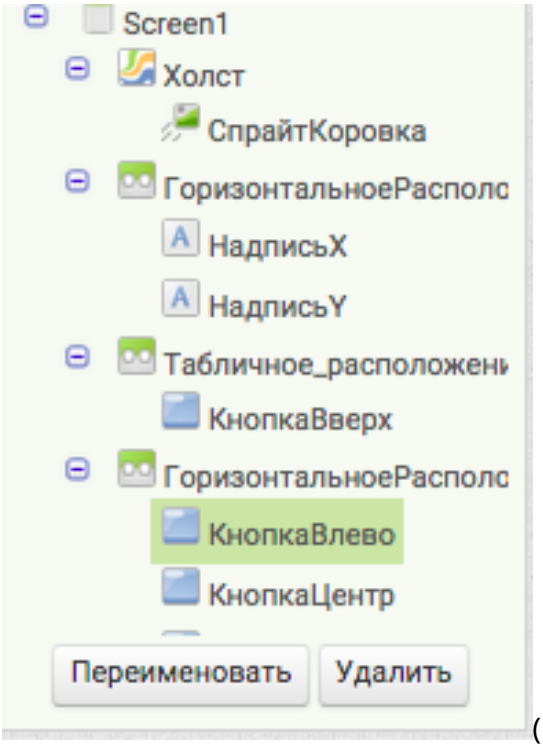
1. **Встроенные блоки**

Данная группа блоков позволяет задавать определенные действия/функции созданным компонентам.



- **Управление—** содержит общие для всех компонент блоки ветвления, цикла, работы с несколькими экранами и пр.
- **Логика—** содержит блоки для использования логических функций в приложении
- **Математика-** содержит набор математических блоков
- **Текст** - включает набор текстовых блоков
- **Массивы** - содержит блоки для работы с массивами/списками
- **Цвета** - определяет блоки по работе с цветами
- **Переменные** - блоки позволяющие определять и устанавливать значение глобальных и локальных переменных
- **Процедуры** - содержит блоки, позволяющие определять процедуры и функции, с параметрами или без них, внутри приложения

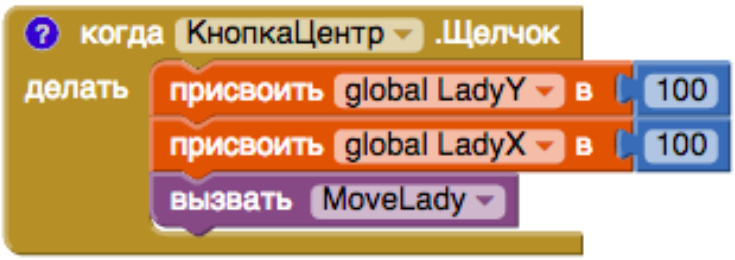
- **Блоки действий/событий для компонентов вашего приложения (Группа Screen 1)**. Задает действия компонентам конкретного приложения. При выделении нужного компонента, отображаются доступные для него блоки.



(

- Любой компонент. Данная группа блоков позволяет организовать и управлять в приложении большим количеством однотипных компонент, например 20 спрайтами или 40 кнопками.

Конструкции из блоков собираются в поле Просмотр.

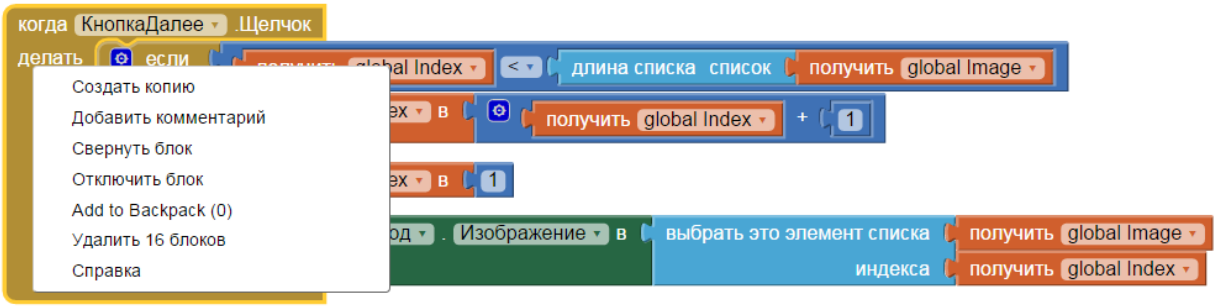


1.5 Функции режима “Блоки”

При работе в режиме “Блоки” часто используются следующие его функции:

- Свернуть/Развернуть блок.

Функция “Свернуть блок” используется для оптимизации места на экране, при создании приложений с большим программным кодом.



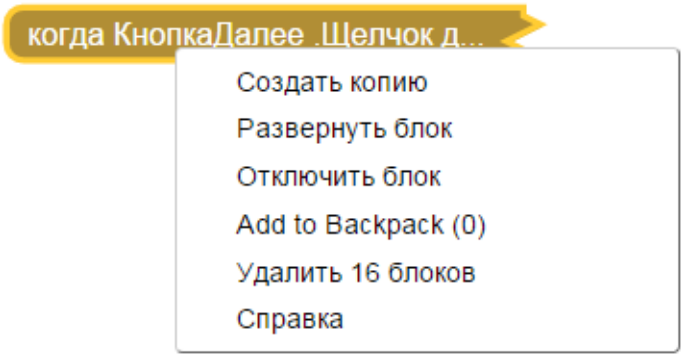
После выполнения функции “Свернуть блок”, конструкция блоков принимает следующий вид:

Чтобы развернуть блок, необходимо щелкнуть по конструкции правой кнопкой мыши и выбрать меню “Развернуть блок”

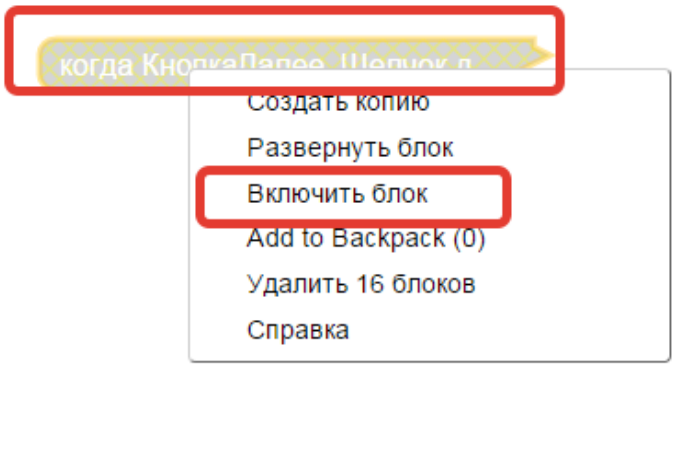
- Добавить комментарий

Комментирование блоков полезно при написании любых программ, вы оставляете комментарий для пояснения действий и событий, которые заложены в этой конструкции.

- Отключить/Включить блок



Данная функция может использоваться при тестировании программ, чтобы не удалять блоки, когда вы сомневаетесь в правильности их использования. Вместо удаления в корзину, на время можно отключить их использование.



- Удалить блоки

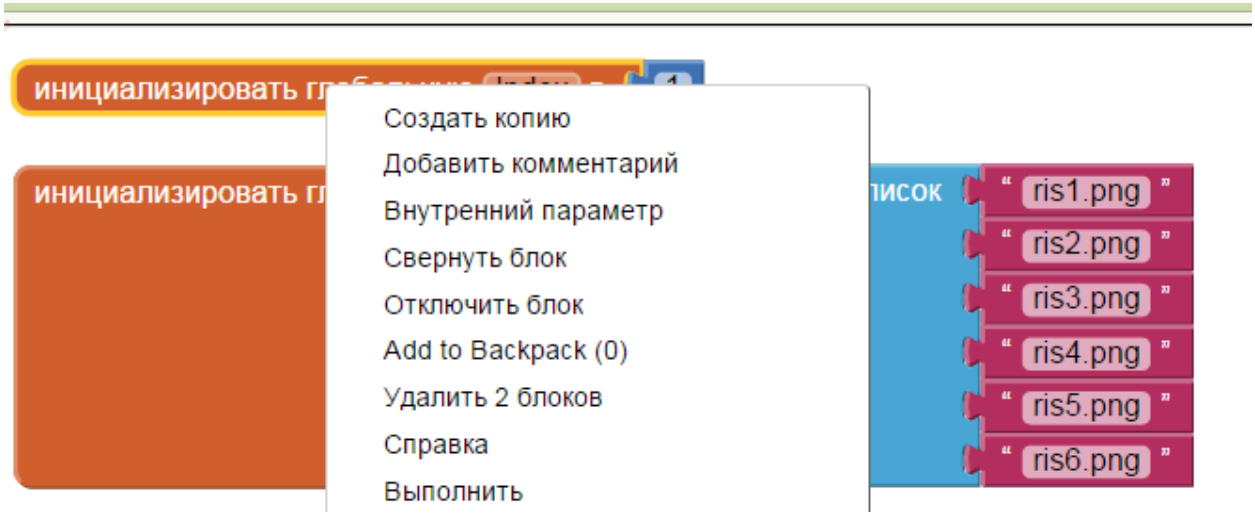
Блоки удаляются без перетаскивания в корзину

- Выполнить

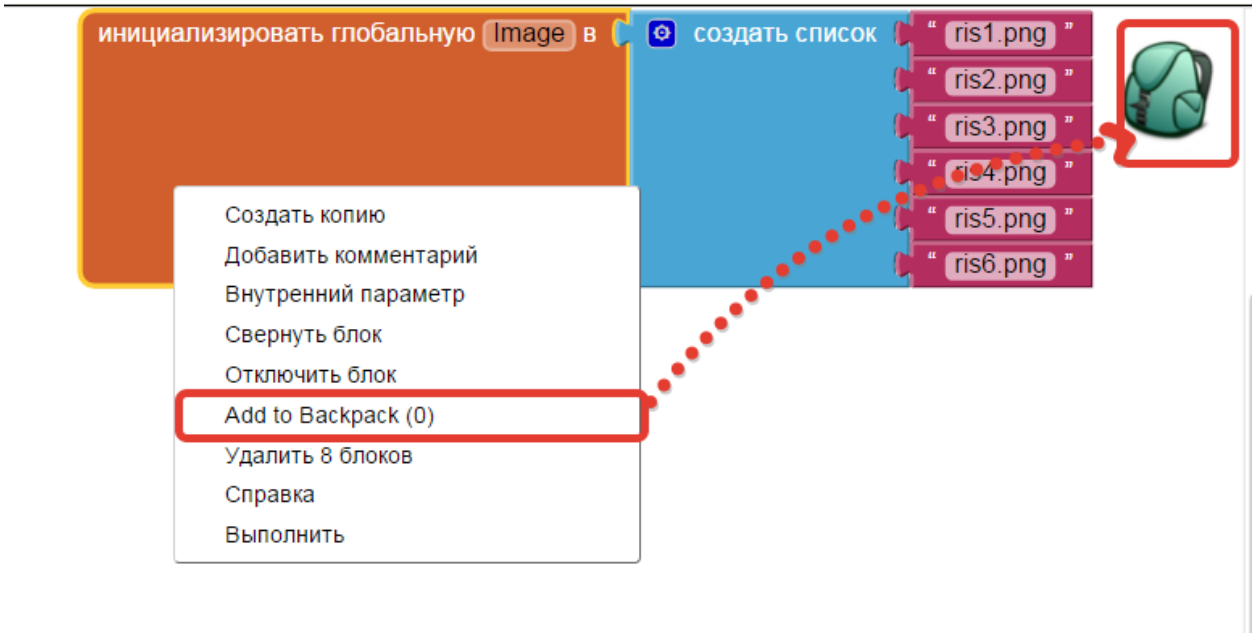
Функция позволяет запустить к исполнению любую часть кода и используется при тестировании программ. В этом случае необходимо иметь подключение к эмулятору.

- Копирование блоков

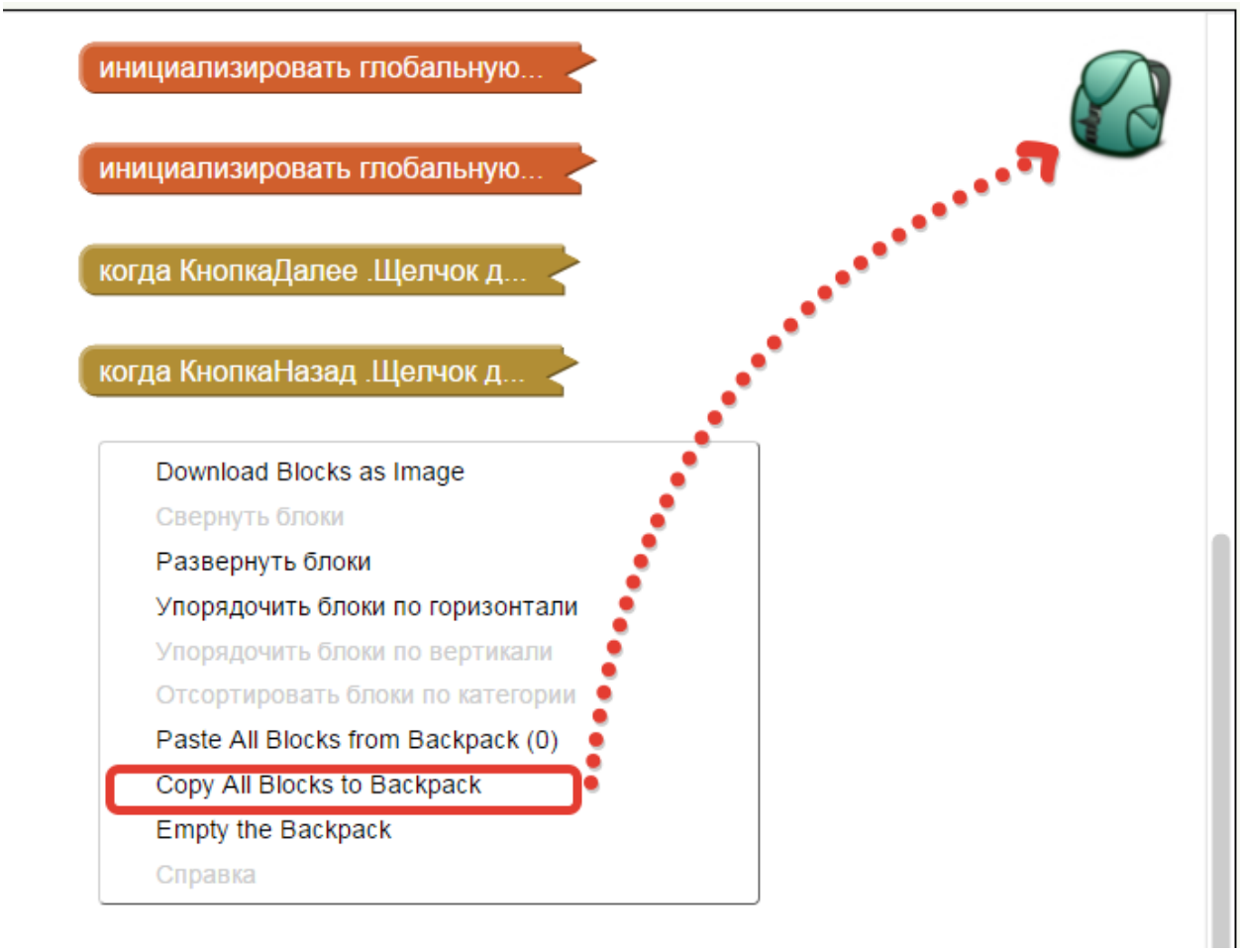
Копировать блоков внутри экрана, можно используя контекстное меню, Для этого необходимо щелкнуть правой кнопкой мыши на нужной конструкции блоков и выбрать “Создать копию”



- Копировать блок в рюкзак



- Копировать все блоки в рюкзак

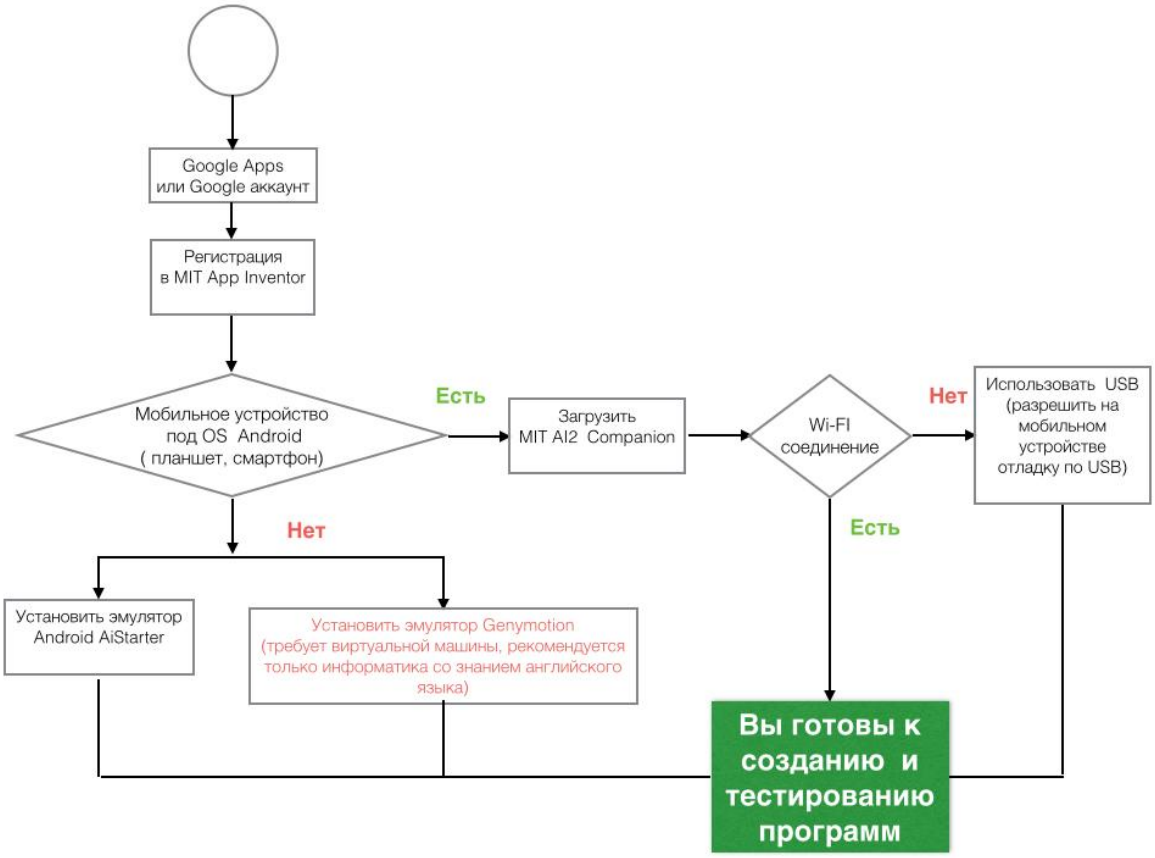


1.6 Загрузка и установка приложения на устройство

Разработка приложения происходит в облачной среде MIT App Inventor.

Тестирование и отладка происходит на мобильном устройстве.

Рекомендуется для разработки использовать настольный ПК или ноутбук, а для отладки и тестирования -мобильное устройство с предустановленным приложением MIT App Inventor Companion, которое позволяет считывать QR код созданного вами мобильного приложения для установки его на ваше устройство.



Способы загрузки приложения на устройство:

- в исходном коде (файл с расширением .aia)

Исходный код в формате .aia позволяет редактировать приложение. Исходный код генерируется со страницы проекта меню Проекты / Экспортировать выбранные проекты (.aia) на Мой компьютер.

- в виде исполняемого файла (файл с расширением .apk)

Файл приложения .apk генерируется в App Inventor в меню Построить - Приложение (сохранить .apk на компьютер). Файл .apk является исполняемым приложением, которое работает на устройстве.

- в виде QR- кода приложения

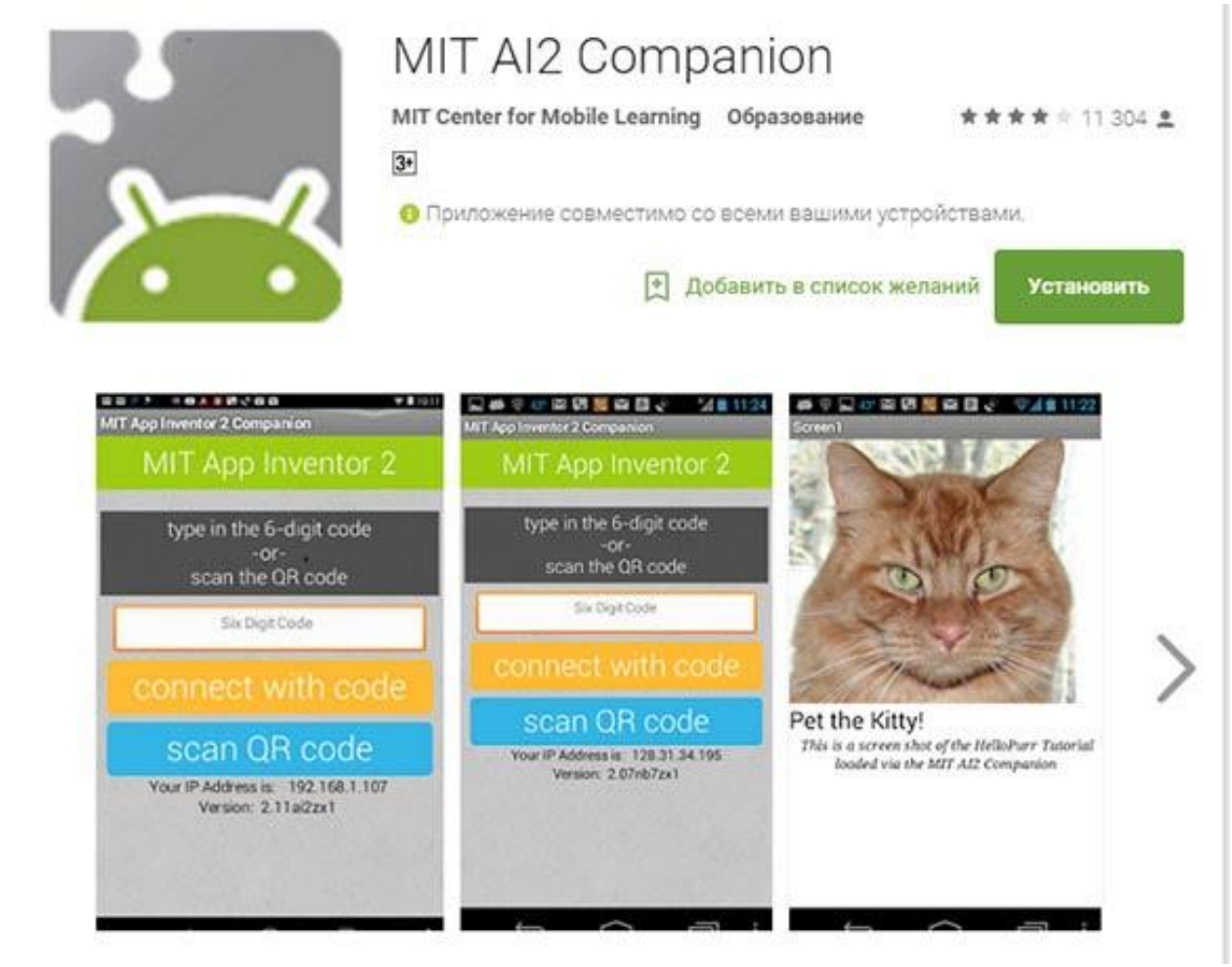
Генерируется с помощью команды меню Построить - Приложение (создать QR код для скачивания .apk).

Для считывания QR кода и установки приложения на мобильное устройство необходимо установить приложение MIT AI2 Companion App из Google Play. на мобильное устройство.

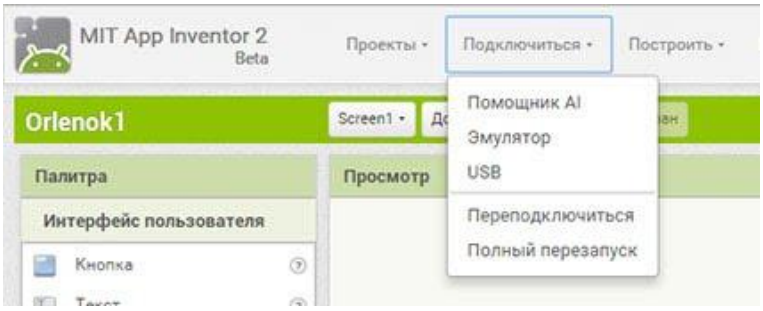
При установке ваших приложений .apk на мобильное устройство, необходимо разрешить установку приложений из неизвестных источников (Настройки-Приложения-Неизвестные источники).

1.6.1 Если у вас есть мобильное устройство с OS Android и Wi-Fi соединение

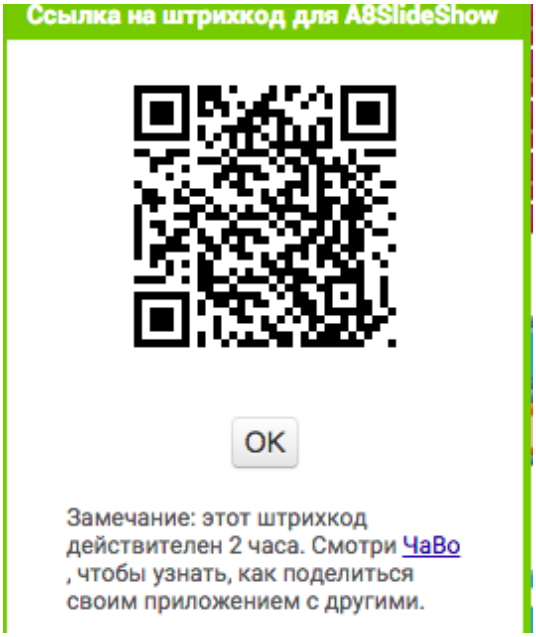
1. На мобильном устройстве загрузить и установить из магазина Google Play приложение **MIT AI2 Companion App**.



2. Подсоединить компьютер на котором вы работаете и мобильное устройство к сети с доступом к интернет, например, через Wi-Fi.
3. На компьютере открыть проект, который нужно протестировать, и выбрать в меню **Помощник->Помощник AI.**



4. На экране компьютера появится QR код вашего приложения.



5 Запустить MIT Ai2 Companion на мобильном устройстве и нажать **Scan QR code**. Через несколько секунд приложение появится на вашем устройстве.

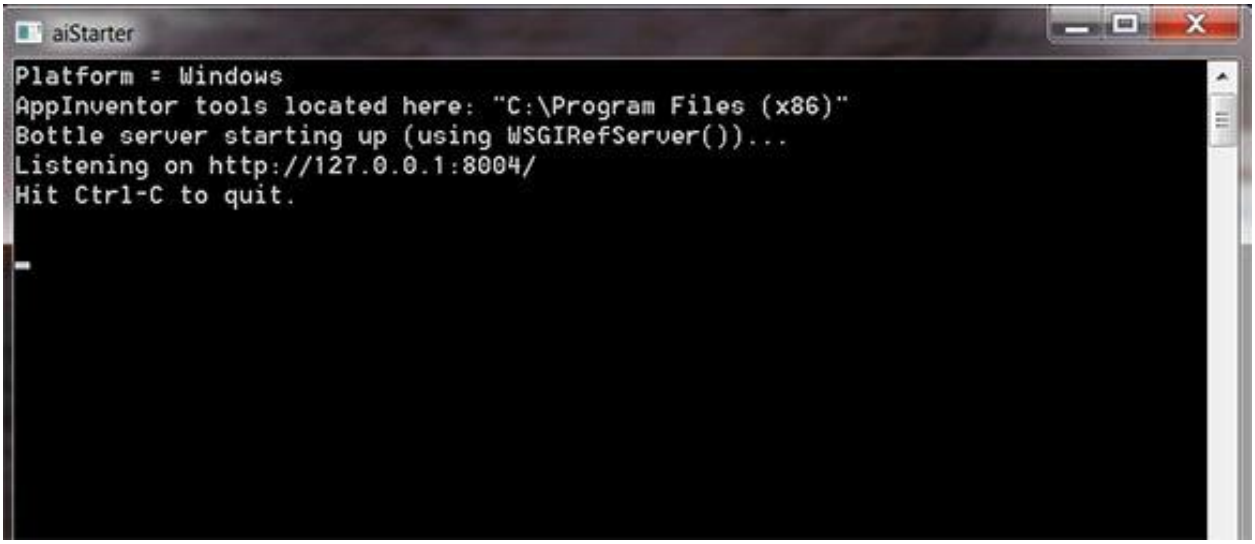
1.6.2 Если у Вас отсутствует мобильное устройство с OS Android?

1. Загрузить и установить специальное программное обеспечение **App Inventor Setup Software**

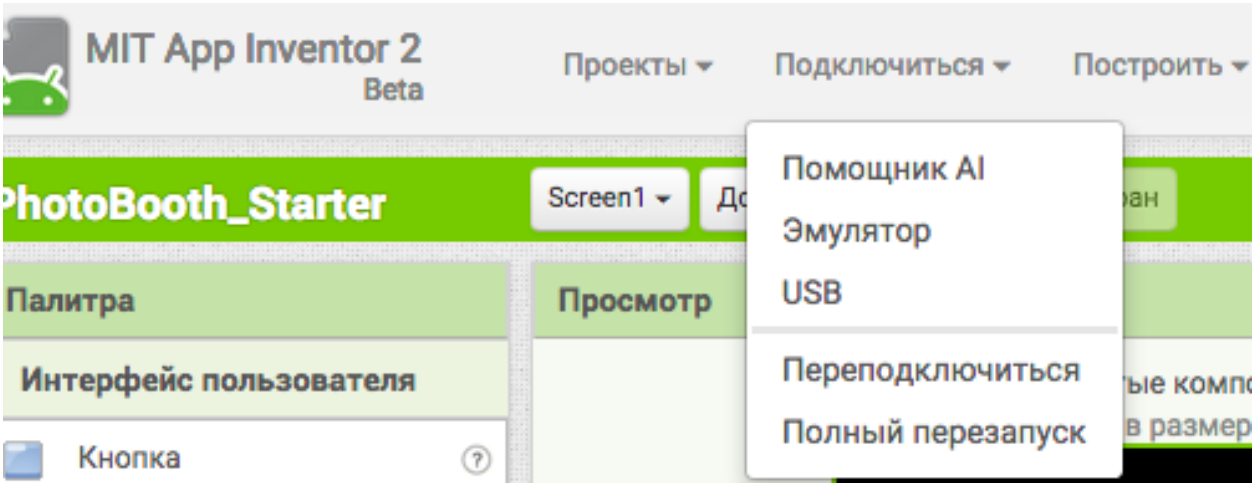
- Для Mac OS X (англ. Instructions for Mac OS X)
- Для ОС Windows (англ. Instructions for Windows)
- Для Linux (англ. Instructions for GNU/Linux)

2. Запустить aiStarter (только для Windows & GNU/Linux)

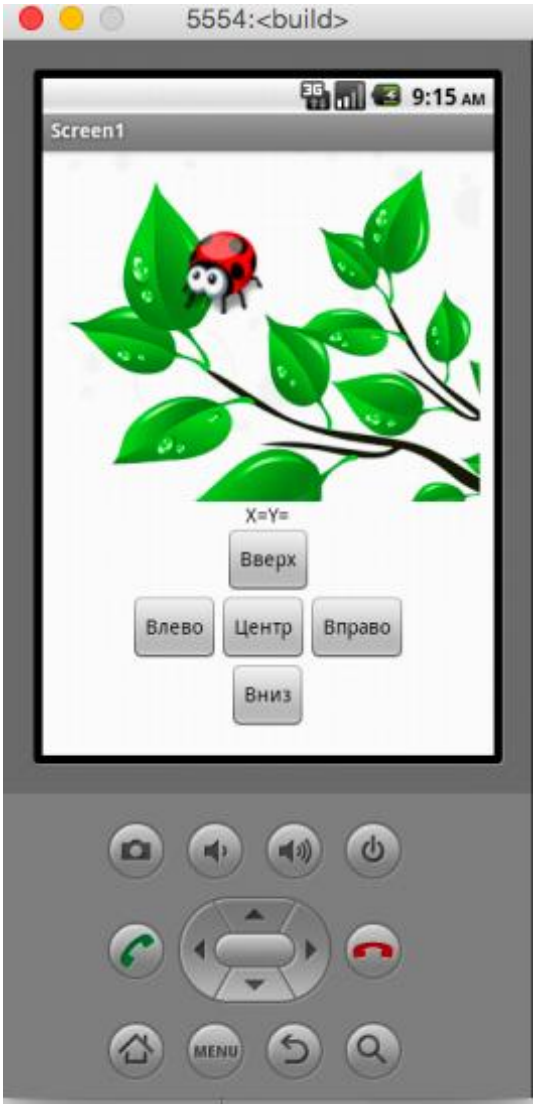
aiStarter будет успешно запущен, если отображается окно следующего вида:



3. Перейти в окно проекта в MiT App Inventor и выбрать меню **Подключиться -> Эмулятор.**

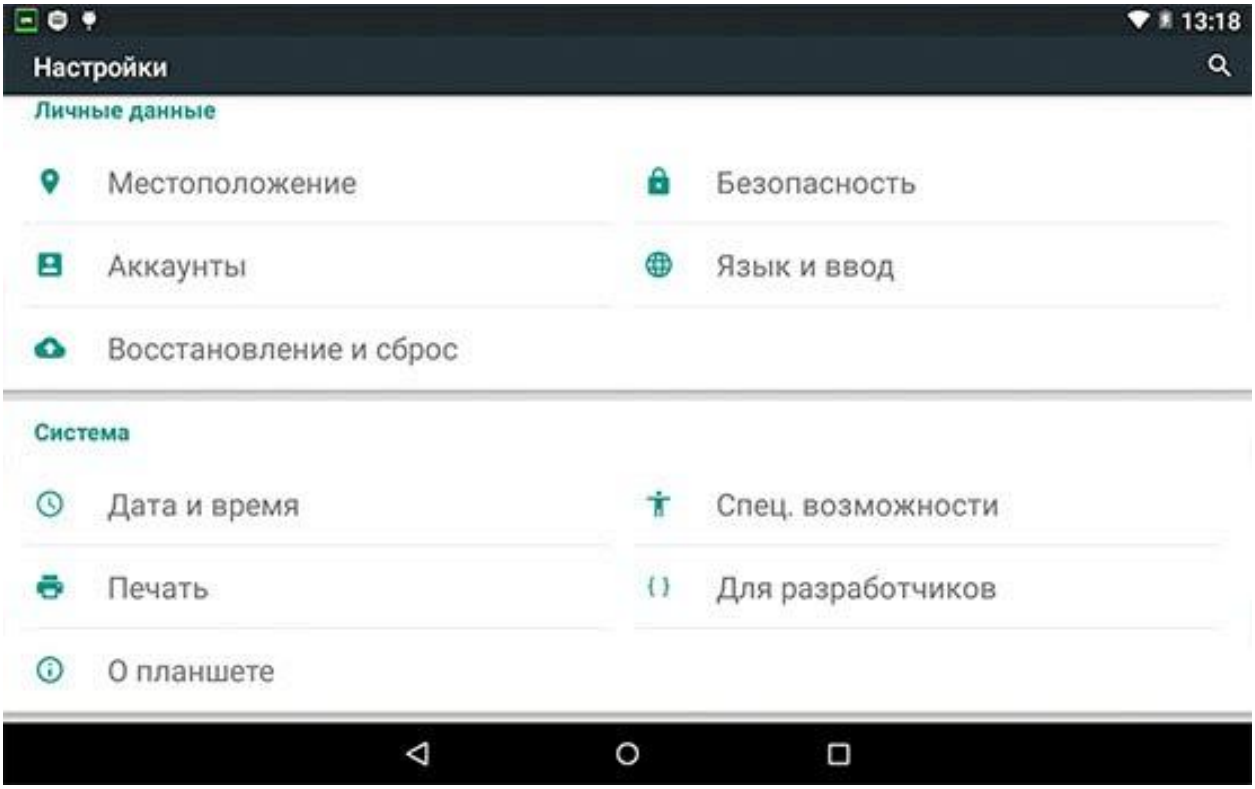


Окно эмулятора имеет следующий вид:

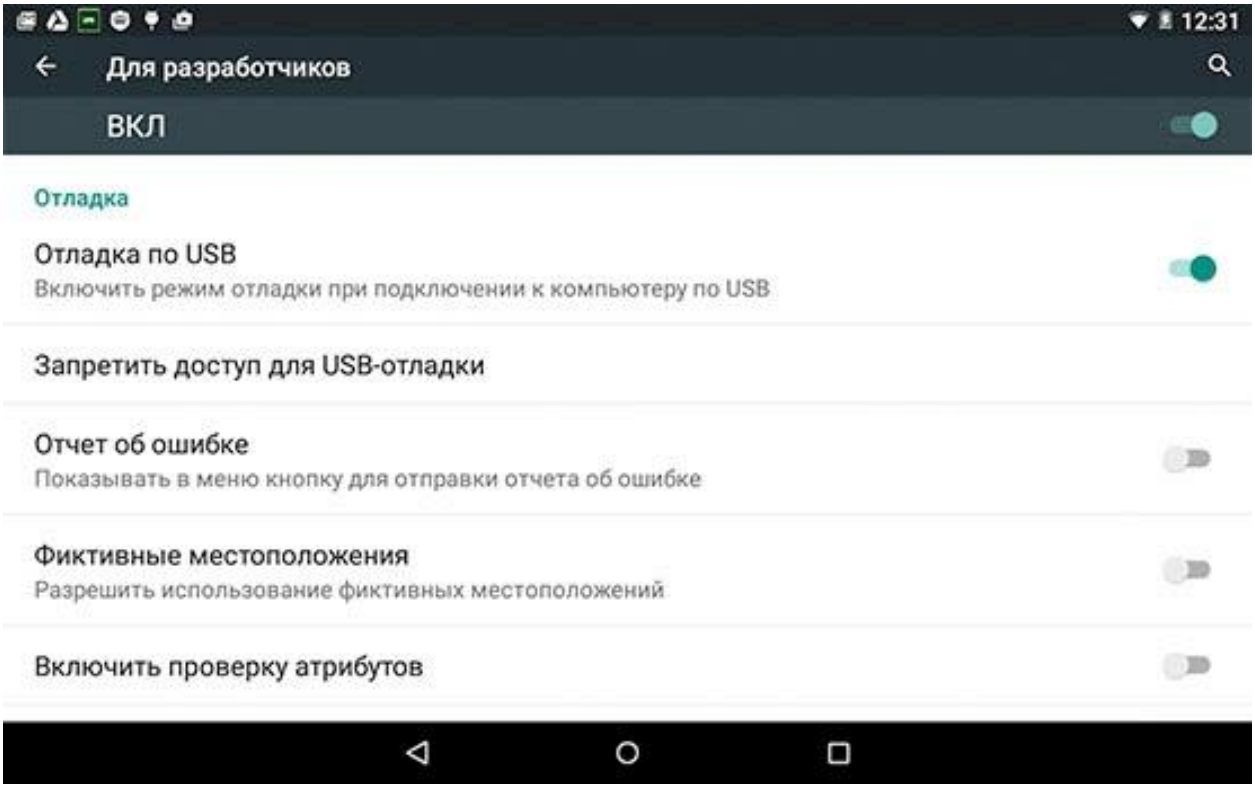


1.6.3 Если вы используете USB кабель

1. Подготовить устройство для использования USB (Включить отладку по USB).



На Android устройстве, перейти в меню **Настройки системы-> Для разработчиков**, и включить пункт меню Отладка USB.

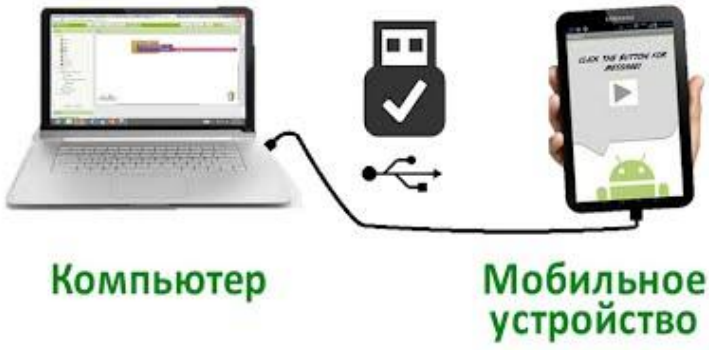


На большинстве устройств, работающих под управлением Android 3.2 или старше, выбрать опцию в **Настройки-> Приложения-> Разработка**.

На Android 4.0 и новее, это в **Настройки-> Функции для разработчиков**. На Android 4.2 и старше, Функция для разработчиков по умолчанию скрыта. Чтобы включить данную функцию, перейдите в меню Настройки->О телефоне и нажмите номер сборки семь раз. Далее необходимо вернуться к предыдущему экрану, чтобы найти меню Для разработчиков, в том числе "USB Debugging".

2. Подключить мобильное устройство к компьютеру

Подключить Android устройство к компьютеру с помощью кабеля USB, убедиться, что устройство подключается как "запоминающее устройство",а не как "медиа-устройство".



На Android 4.2.2 и старше, при первом подключении мобильного устройства к компьютеру появится экран с сообщением **Разрешить USB-отладку**, для подключения его к компьютеру нажмите "OK".

3. Проверить соединение
Убедитесь в том, что ваш компьютер подключен к мобильному устройству.

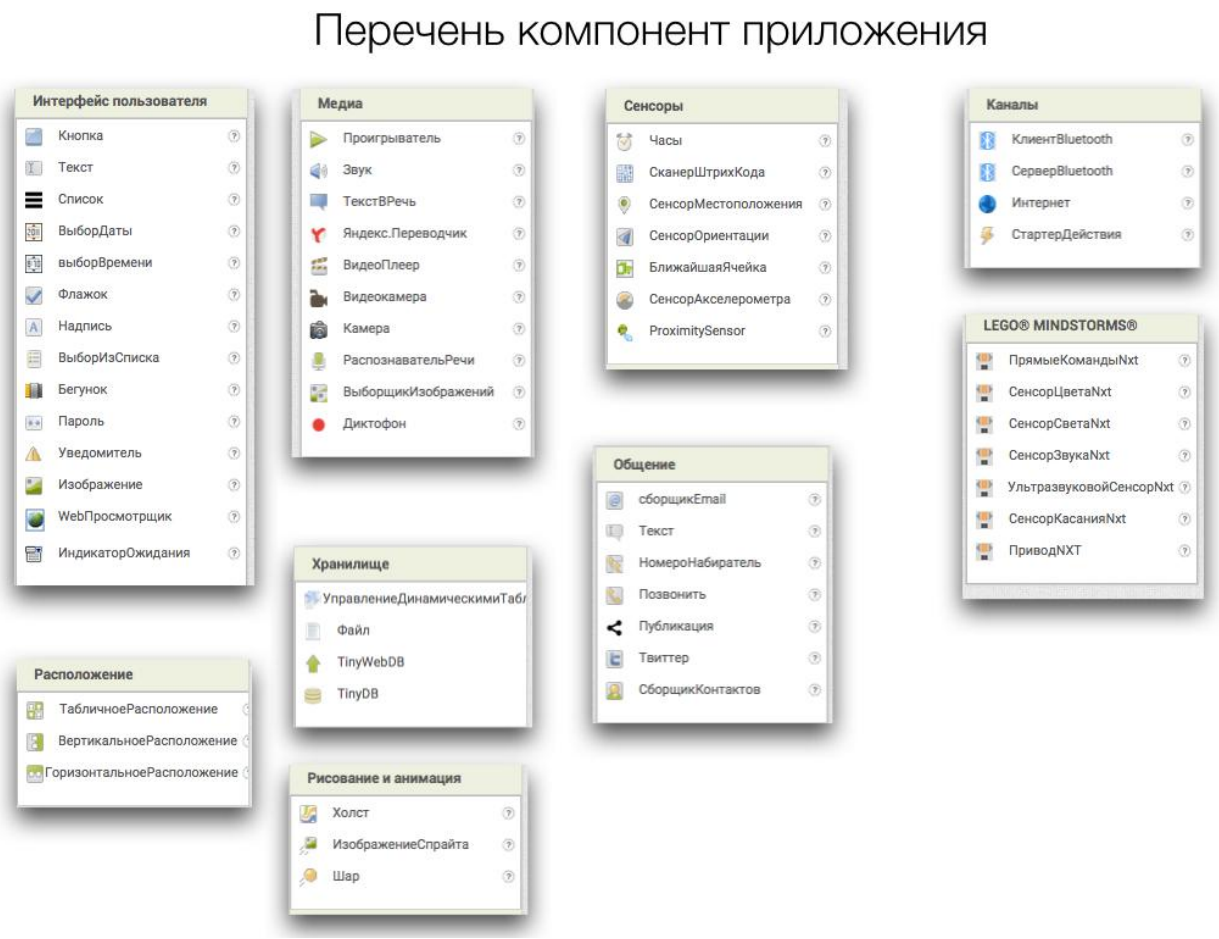
4 . Скопировать файл .APK, сохраненный на вашем компьютере, в папку на мобильном устройстве.

1.7 Загрузка .apk файла на мобильное устройство

Загруженный для установки .apk файл приложения хранится на мобильном устройстве в каталоге Download. Доступ к нему можно получить с помощью приложения “файловый менеджер”.

Для разных OS Andriod версии одного и того же приложения могут перезаписываться в различных вариантах. Новое приложение может быть установлено на место старого, или в виде новой версии с модифицированными именем.
Рекомендуется периодически в каталоге Download удалять старые версии приложения.

1.8 Компоненты приложения



Компоненты приложения размещаются на экране в режиме “Дизайн”. Все компоненты разделены на несколько групп.

- **Интерфейс пользователя** включает такие компоненты как кнопка, текст, флажок, надпись и другие, которые позволяют приложению взаимодействовать с пользователем.

- **Расположение** - компоненты, отвечающие за макетирование экрана, позволяют размещать компоненты интерфейса пользователя горизонтально, вертикально, или в ячейки таблицы. В среде MIT App inventor нет форматирования, позволяющего задавать интервалы между определенными компонентами, поэтому для макета и задания расстояния и пространства между элементами используются компоненты группы расположения с определенными заданными свойствами, например высота или ширина.

- **Медиа** - компоненты, позволяющие задействовать в приложении различные медиа инструменты: устройства, микрофоны и наушники, камеру, звуки и аудиофайлы и другое

- **Рисование и анимация** - группа компонент, позволяющих рисовать или создавать анимацию в приложении.

- **Хранилище** - компоненты, позволяющие передавать значения внутри приложения и сохранять какие -либо данные приложения на внешнем устройстве.

- **Сенсоры** - группа невидимых компонент, позволяющих использовать в приложении данные, полученные с различных сенсоров и датчиков мобильного устройства.

- **Общение** - компоненты, обеспечивающие связь с социальными сетями. позволяющие делиться информацией, получать доступ к контактам устройства и пр.

- **Каналы -** компоненты, позволяющие запустить какое-либо внешнее действие из приложения: другое приложения на мобильном устройстве, камеру, поиск в сети интернет или открыть веб-страницу.

- **LEGO® MINDSTORMS®** - компоненты обеспечивающие управление LEGO® MINDSTORMS® NXT с использованием Bluetooth.

1.9 Разрешение экрана

При работе в среде MIT App Inventor штатное разрешение экрана 320x480 пикселей.
Создание приложения в ней требует установки связи положения и размера каждого компонента с разрешением экрана.

Различные мобильные устройства имеют разные разрешения экранов. Вид приложения на смартфоне и семидюймовом планшете будет выглядеть по разном, и на планшете все компоненты могут быть в 1,5 раза крупнее. При разработке приложений для различных экранов лучший способ задания свойств некоторых компонентов в в процентах от размера экрана.

1.9 Первое мобильное приложение

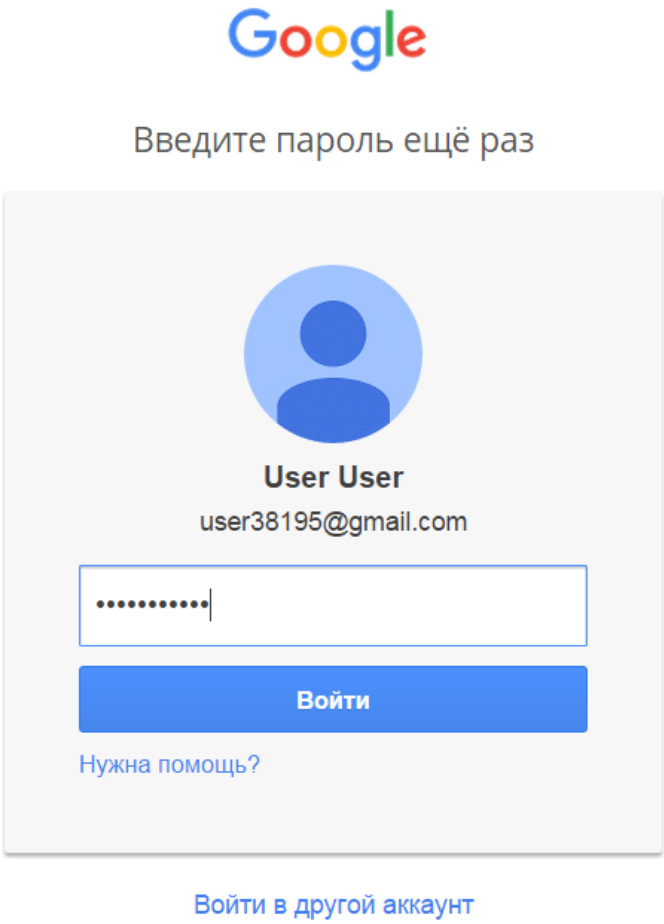
Рассмотрим алгоритм создания приложения на примере проекта, имитирующего игральный кубик, который будет выкидывать кубик с нужным количеством кружков, если мобильное устройство тряхнуть.

Изображения для создания приложения (Скачать)

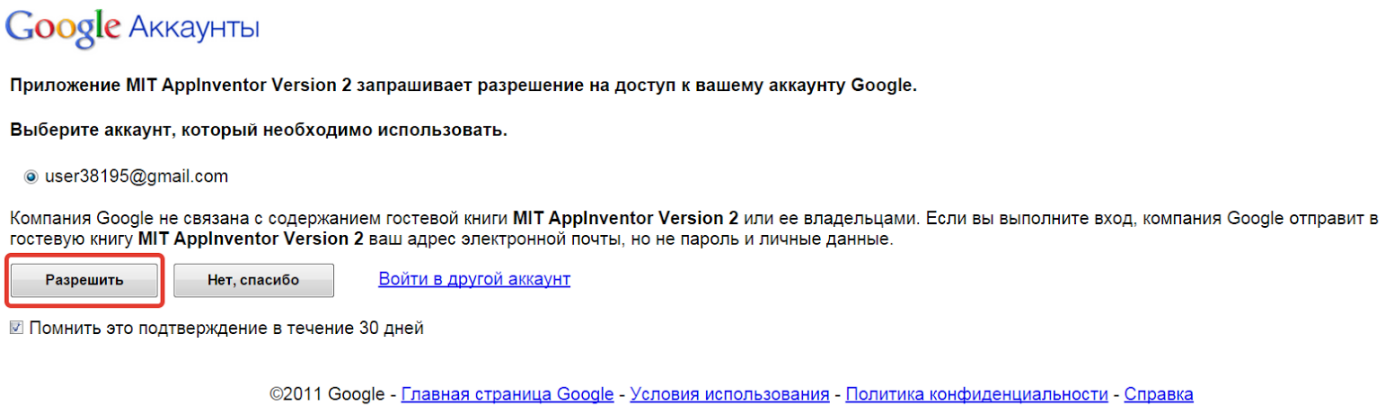
Для выполнения всех шагов алгоритма требуется, чтобы на вашем мобильном устройстве была предустановлено приложение MIT AI2 Companion.

Алгоритм работы:

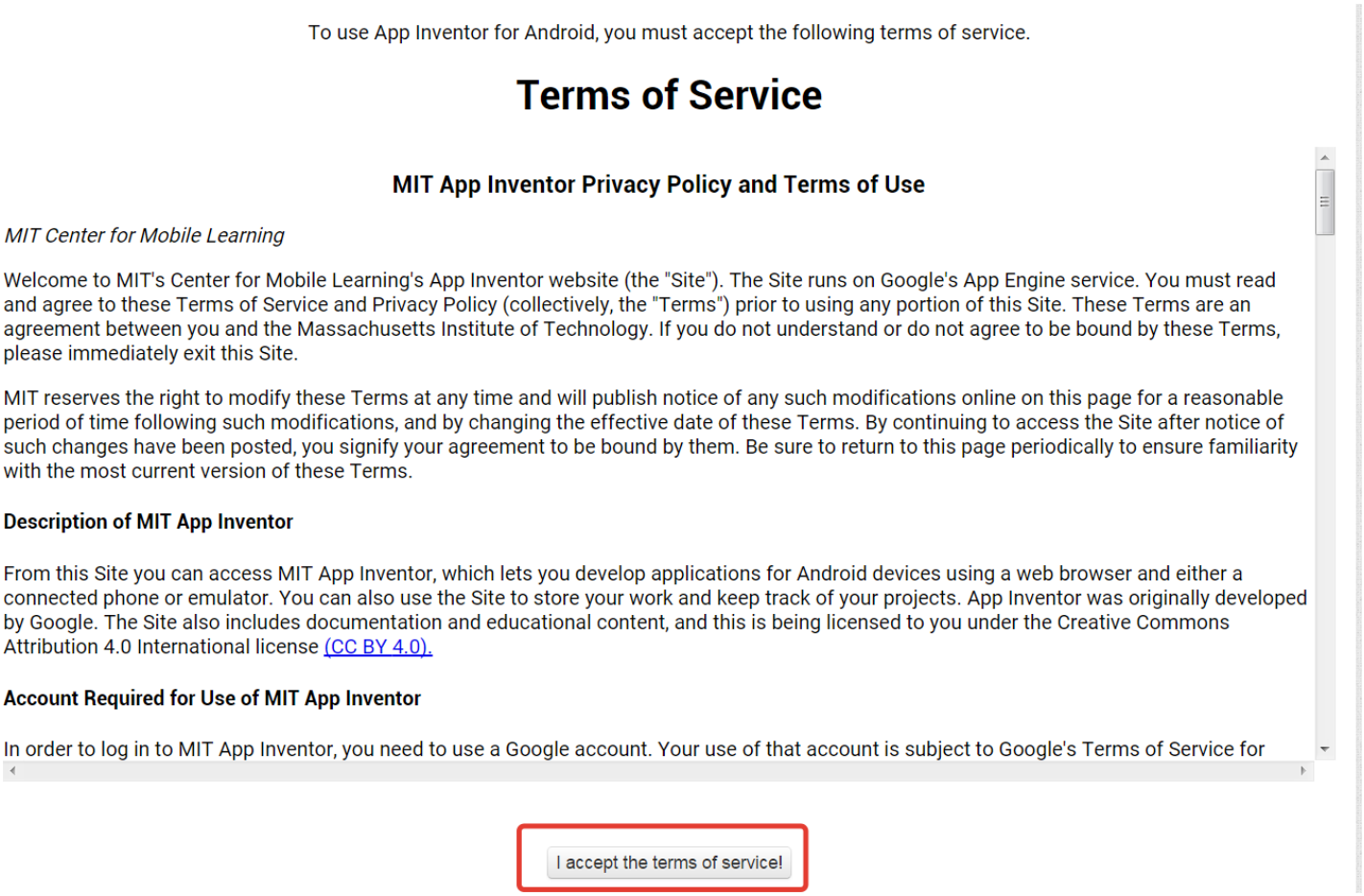
1. Создать Google аккаунт, лучший вариант Google Apps!! или использовать созданный ранее.
2. Войти в среду визуального программирования MIT App Inventor по ссылке http://ai2.appinventor.mit.edu/



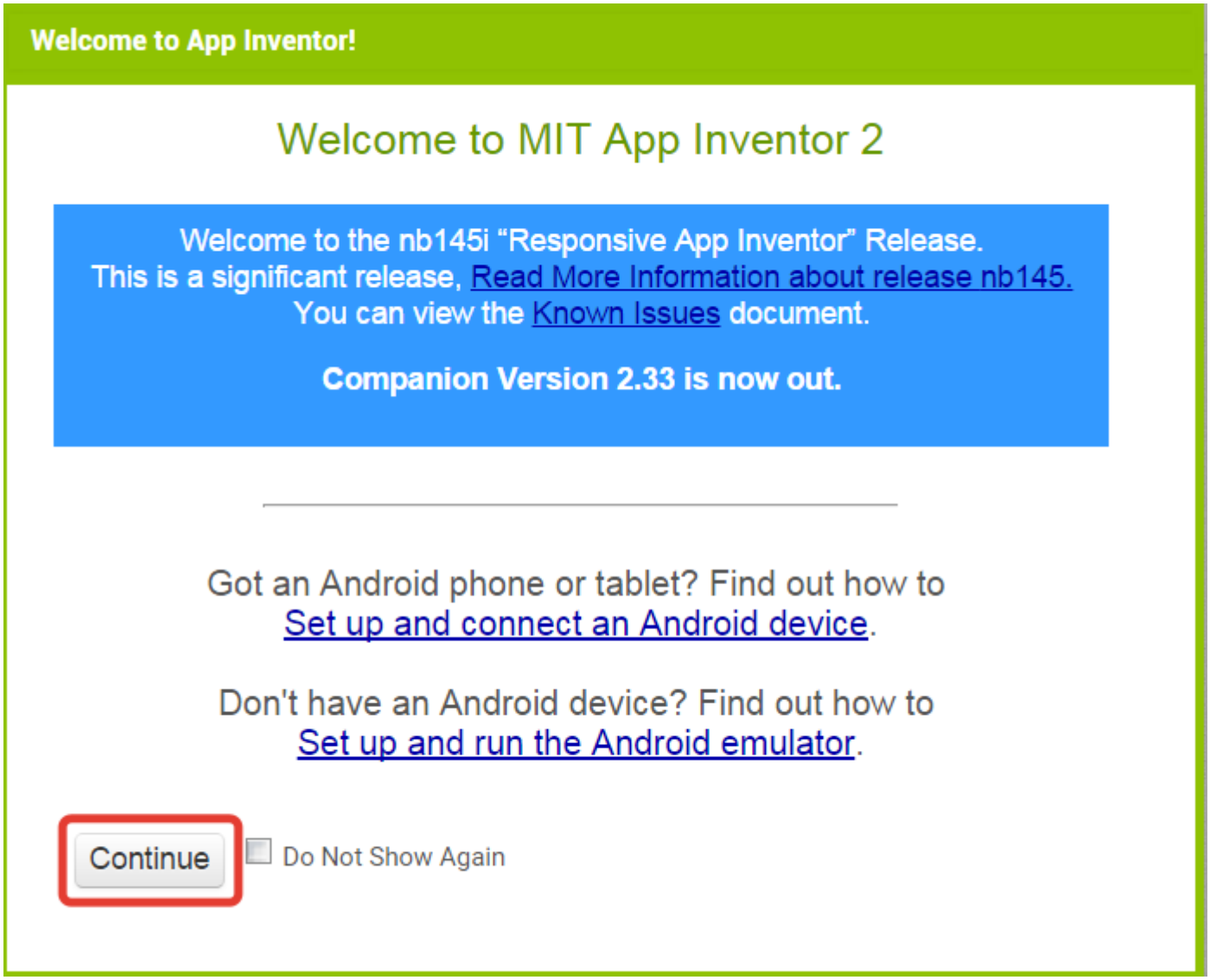
3. Выбрать “Разрешить”



4.В окне “Terms of Service выбрать” I accept the terms of service”

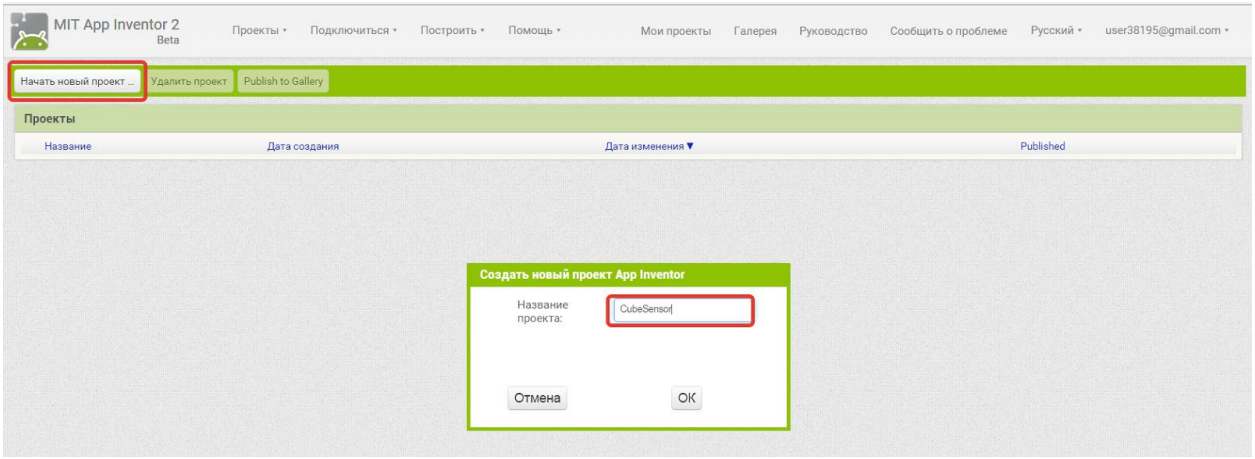


5. В окне “Welcome to App Inventor 2” выбрать “Continue”

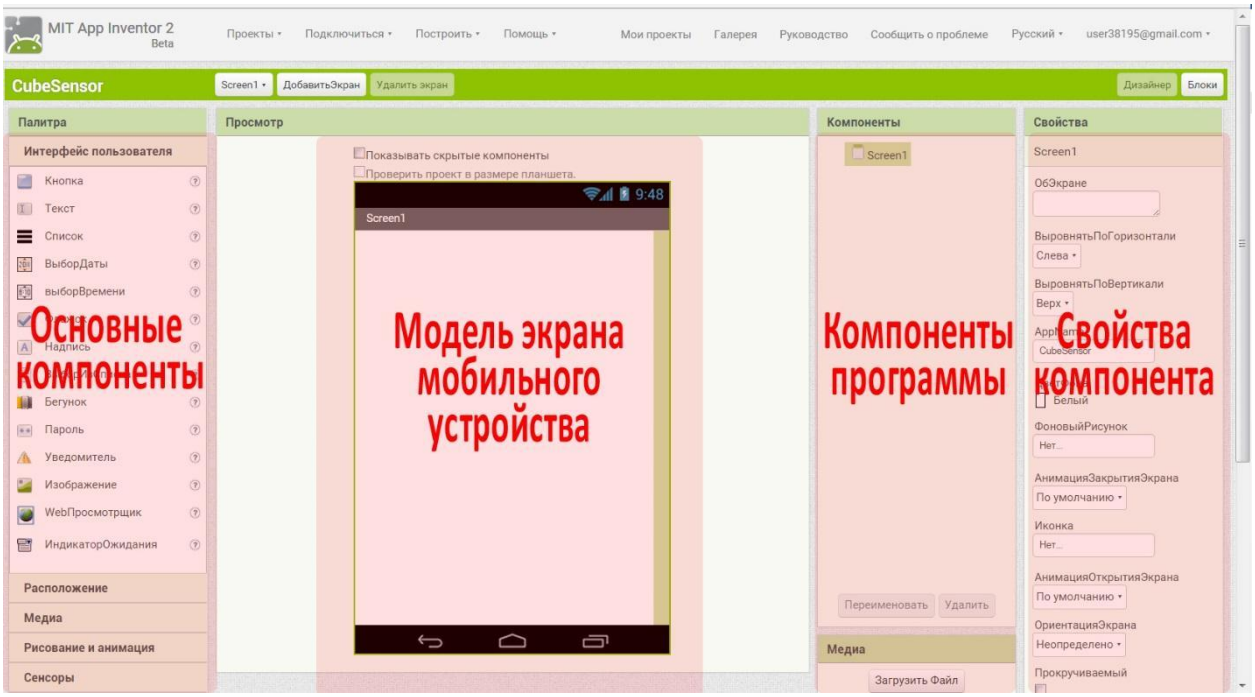


6.Выбрать язык “English-> Русский”

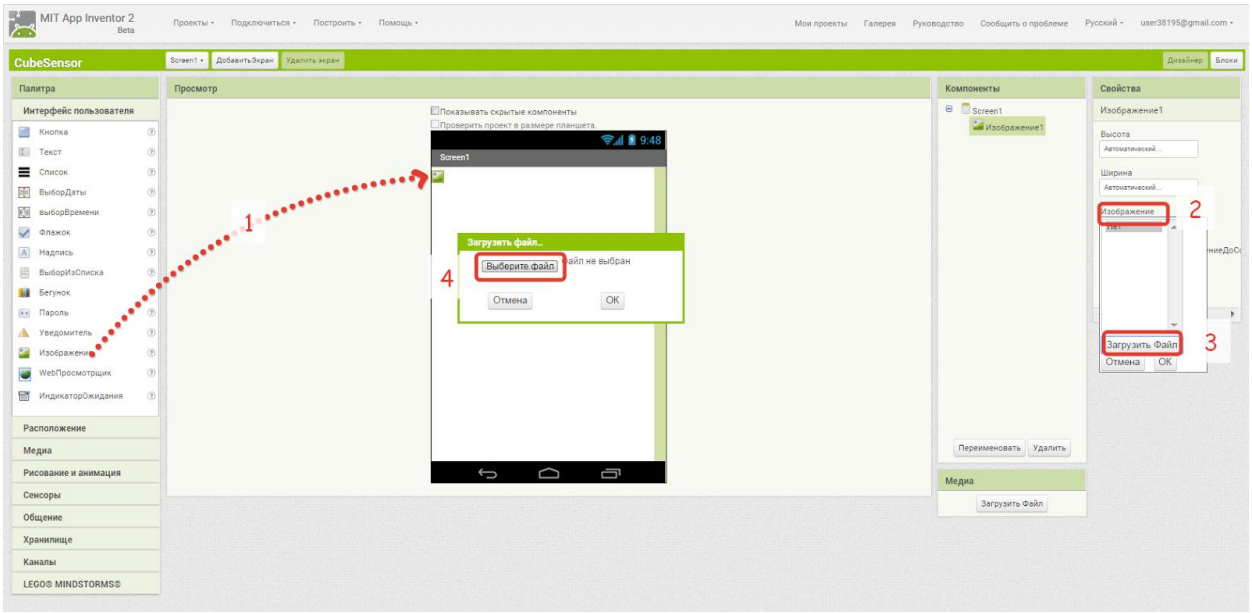
7. Создать новый проект “Начать новый проект->CubeSensor”



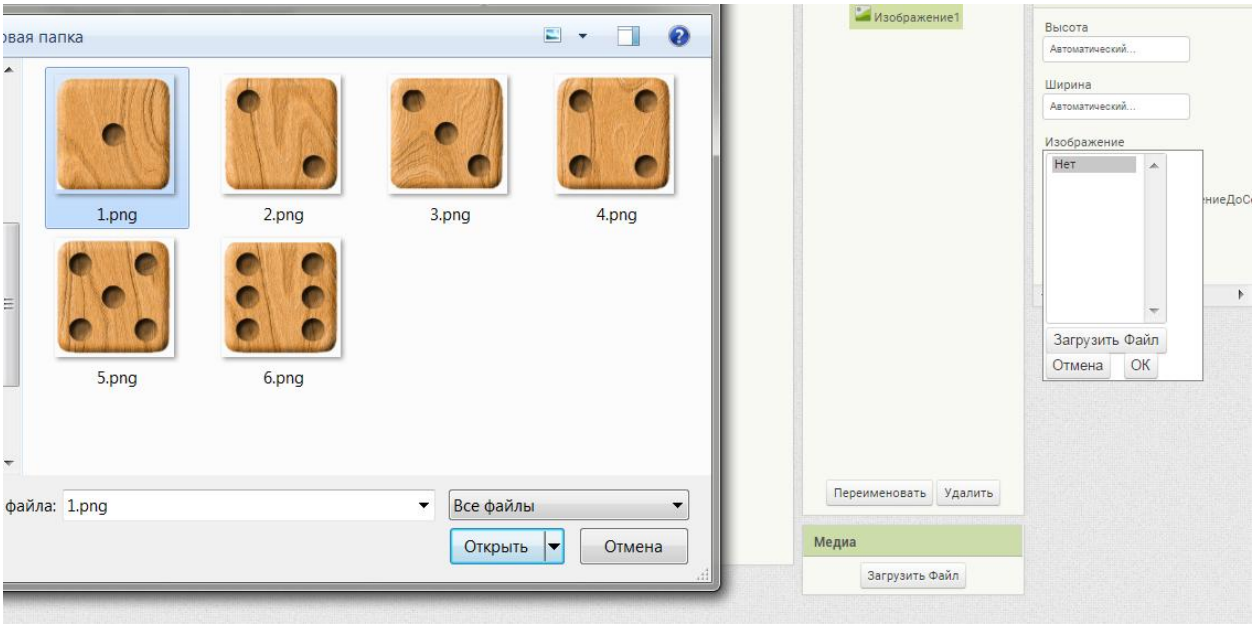
8. Рабочий экран среды визуального программирования будет выглядеть следующим образом:



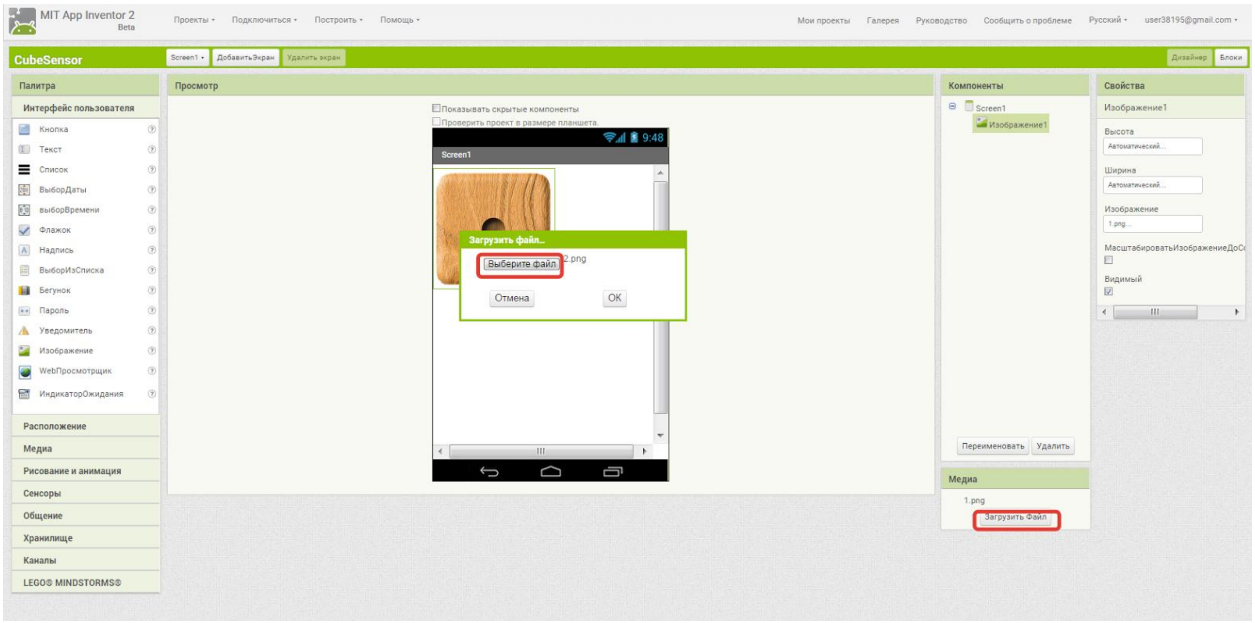
9. Перенести компонент Изображение в окно экрана мобильного устройства, выбрать Изображение->Загрузить в свойствах компонента.



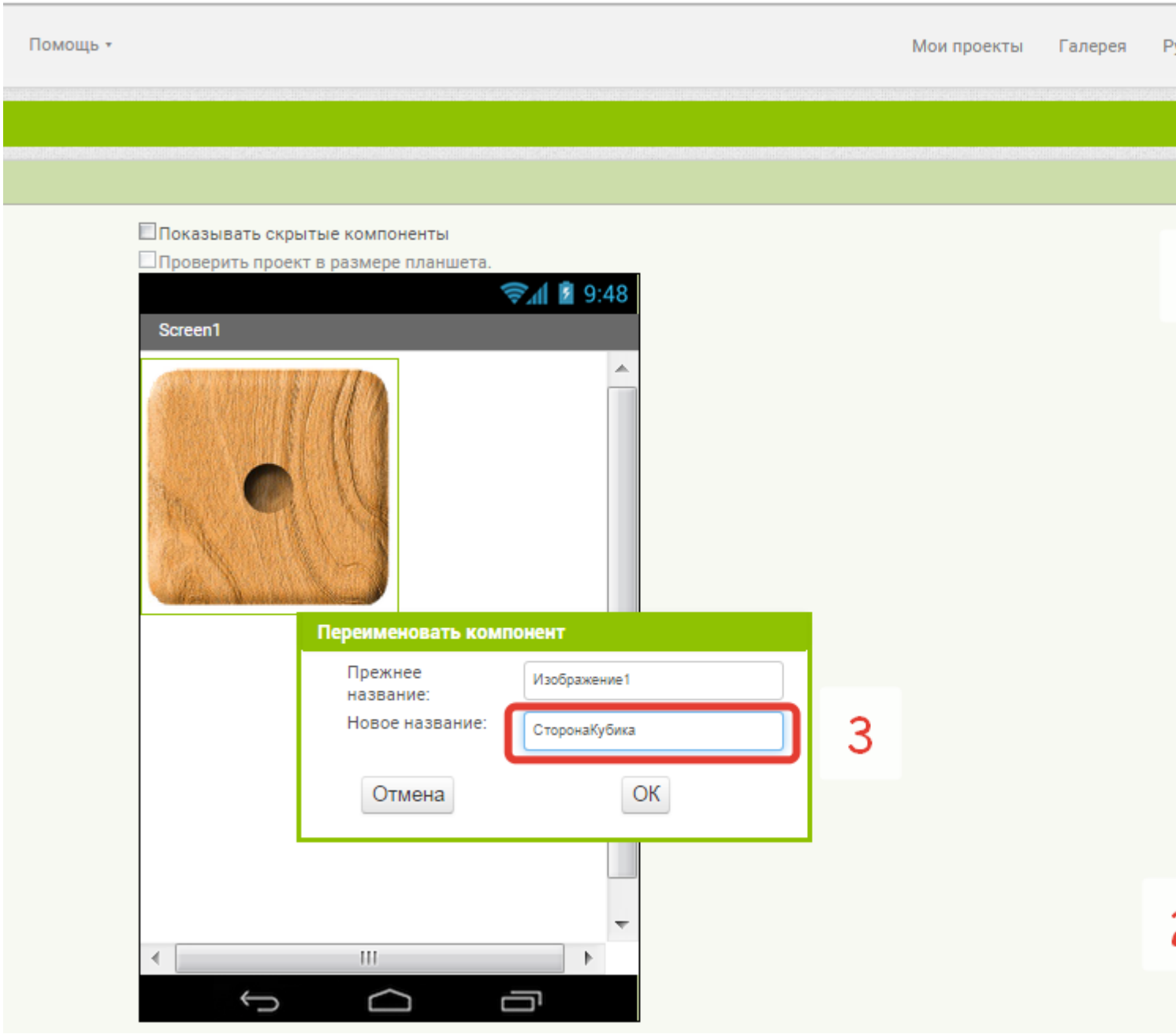
10. Загрузить графический файл для компонента Изображение.



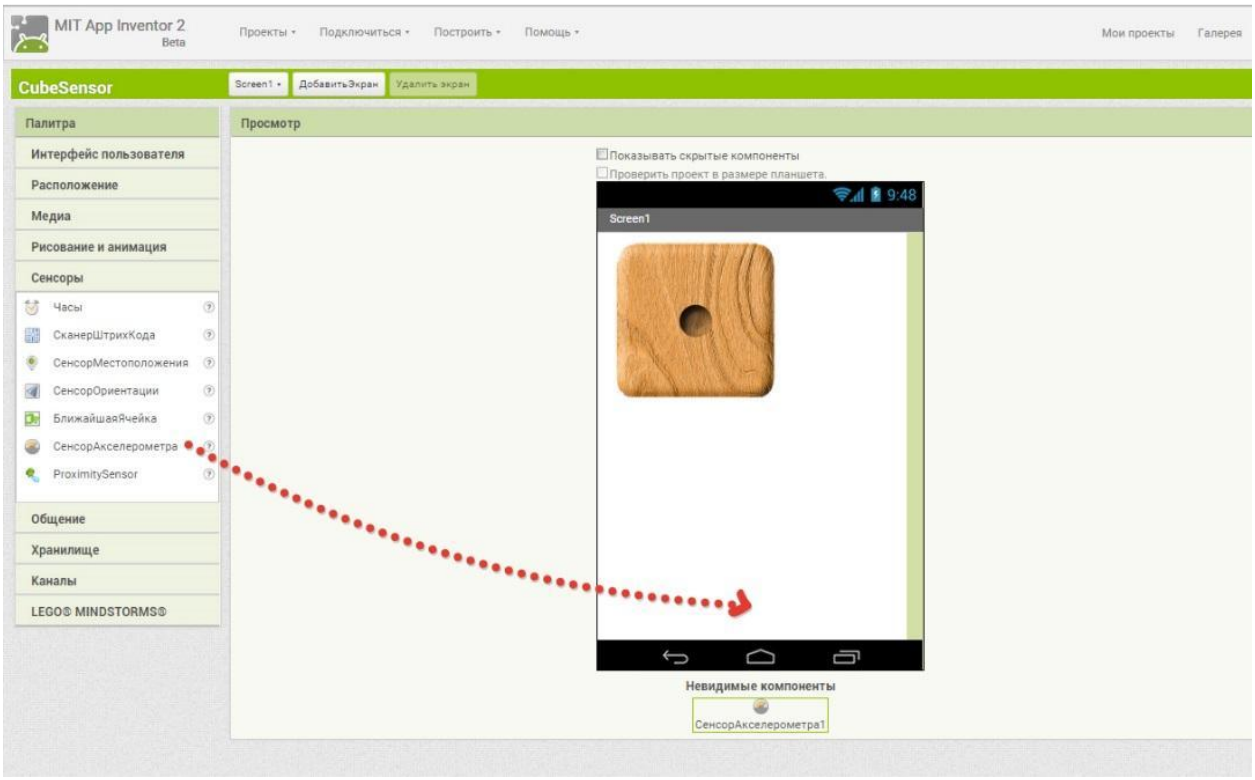
11. Загрузить последовательно 5 графических файлов (сторон кубика) с помощью функции ”Загрузить файл”.



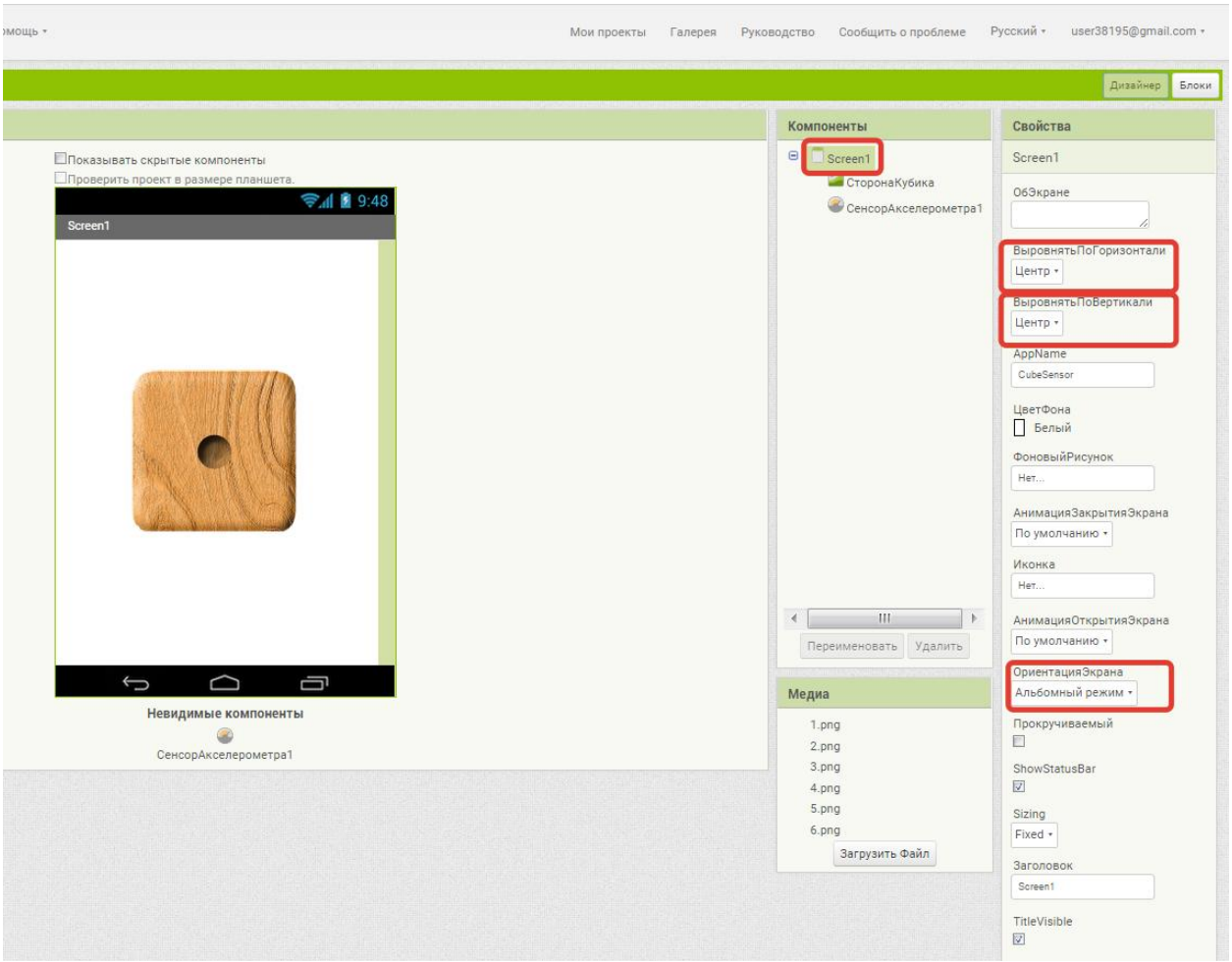
12. Переименовать компонент Изображение1 в СторонаКубика



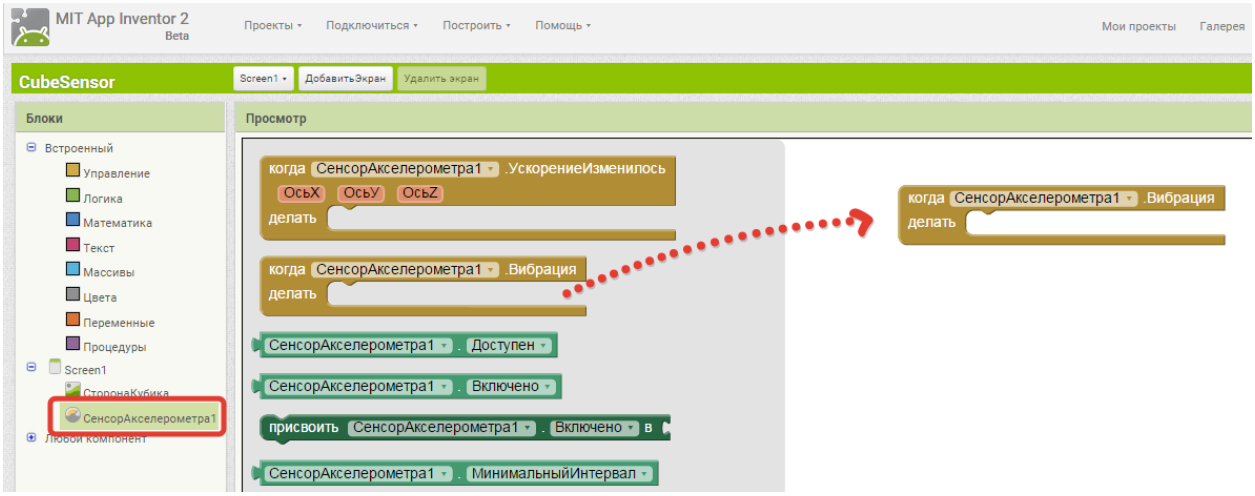
13. Выбрать в группе Сенсоры -> СенсорАкселерометра и перенести его в область экрана мобильного устройства



14. Выбрать компонент **Screen1** и установить его свойства **ВыровнятьПогоризонтали, ВыровнятьпоВертикали, ОриентацияЭкрана**



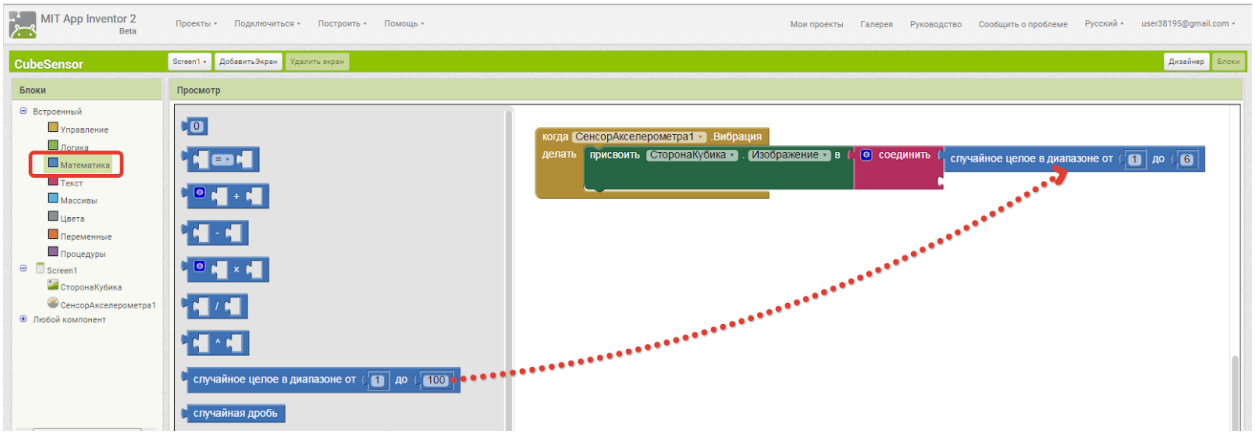
15. Перейти в режим Блоки в меню справа, выбрать **СенсорАкселерометра1** и перетащите блок **когда.СенсорАкселерометра1.Вибрация** в поле блоков программы. Данный блок будет запускаться как только устройство будет подвержено вибрации.



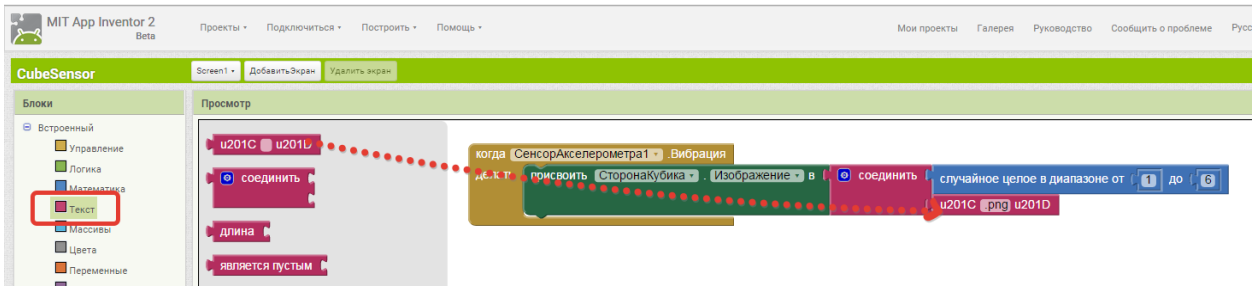
16. Выбрать компонент **СторонаКубика1** и перенести блок **присвоить.СторонаКубика1.изображение** в поле блоков программы. Данный блок выводит изображение графического файла на экран мобильного устройства.

17. Для изображений сторон кубика (файлы 1.png-6.png) имя файла изображения формируется с помощью функции “соединить”: случайное число в диапазоне от 1 до 6 (у нас 6 сторон кубика), плюс расширение графического файла .png.

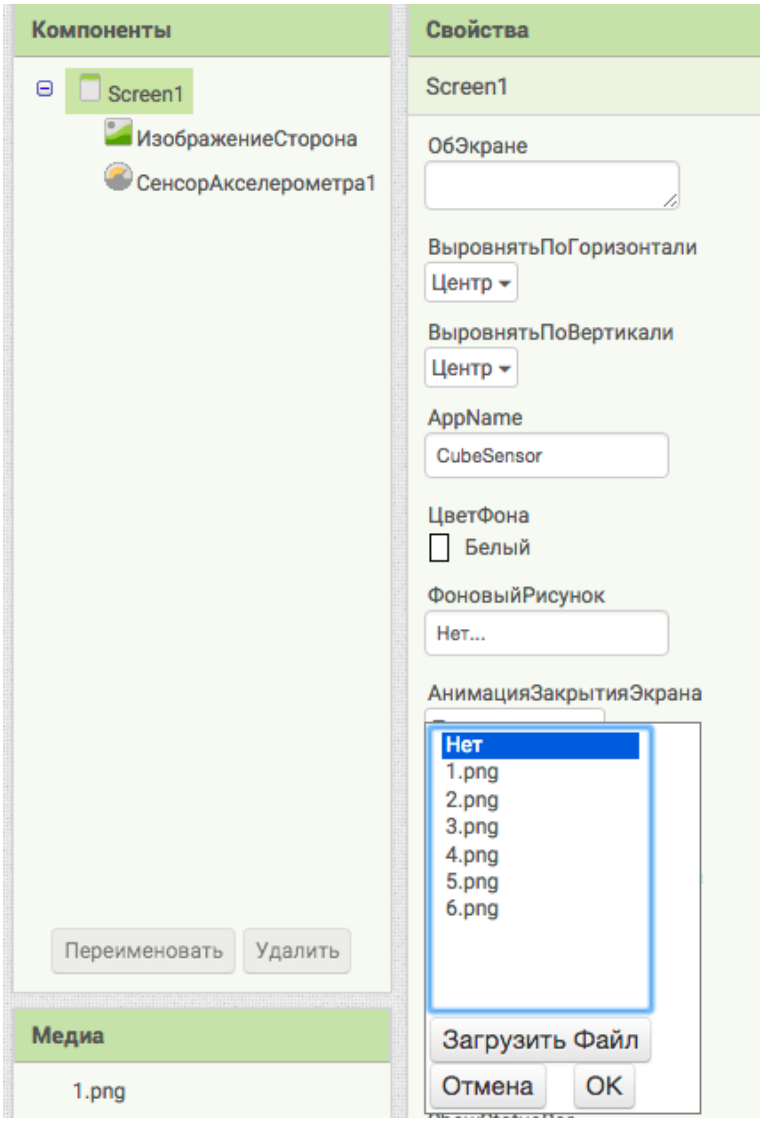
18. Выбрать Математика->Случайное целое от 1 до 100 и установить значения диапазона от 1 до 6.



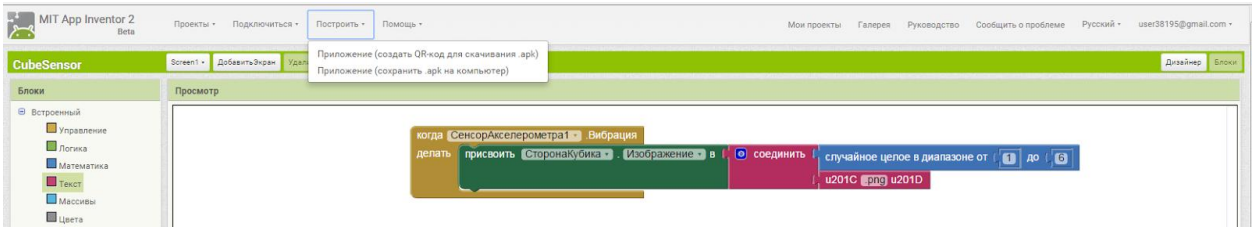
19. Добавить блок  , и вписать туда текст ".png" для последующего соединения со случайным значением от 1 до 6.



20. Оформить приложение и установить иконку в свойствах компонента Screen1.

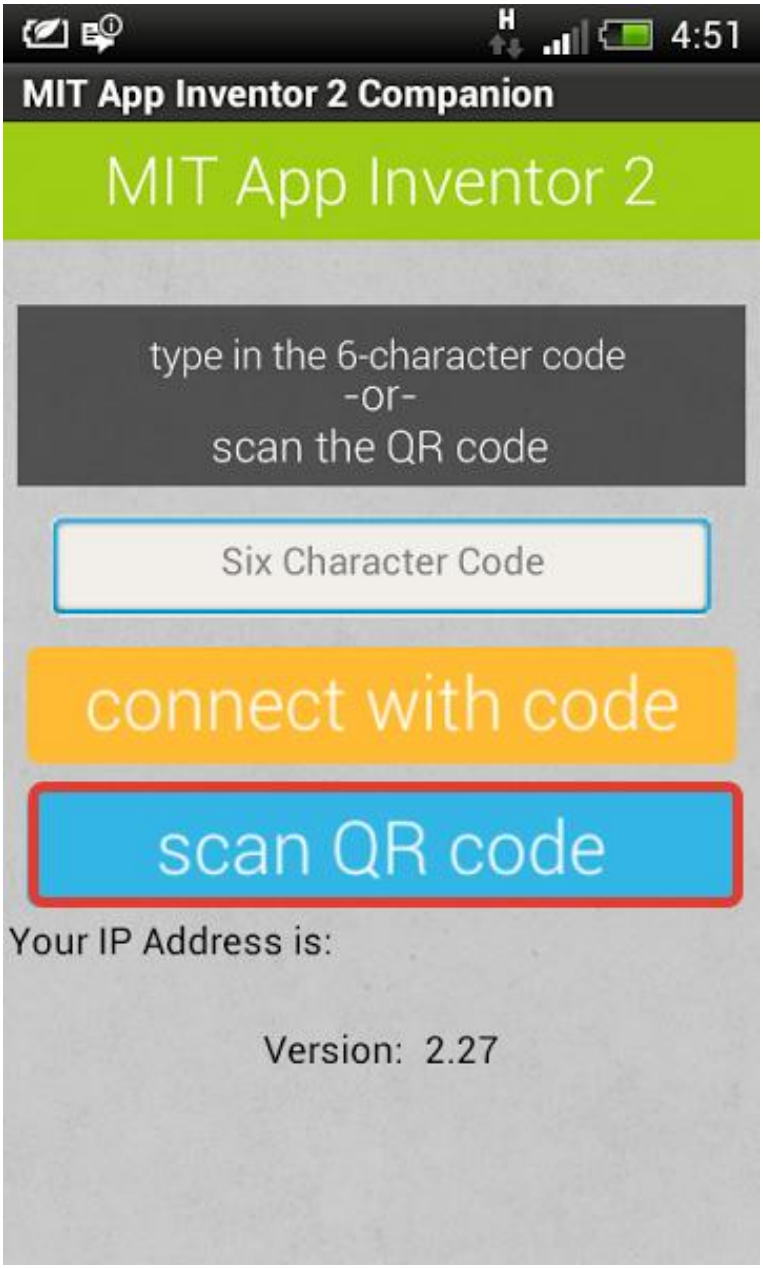


21. Программа готова, необходимо загрузить ее на мобильное устройство для этого нажать Построить->Приложение (Создать QR-код для скачивания .apk)

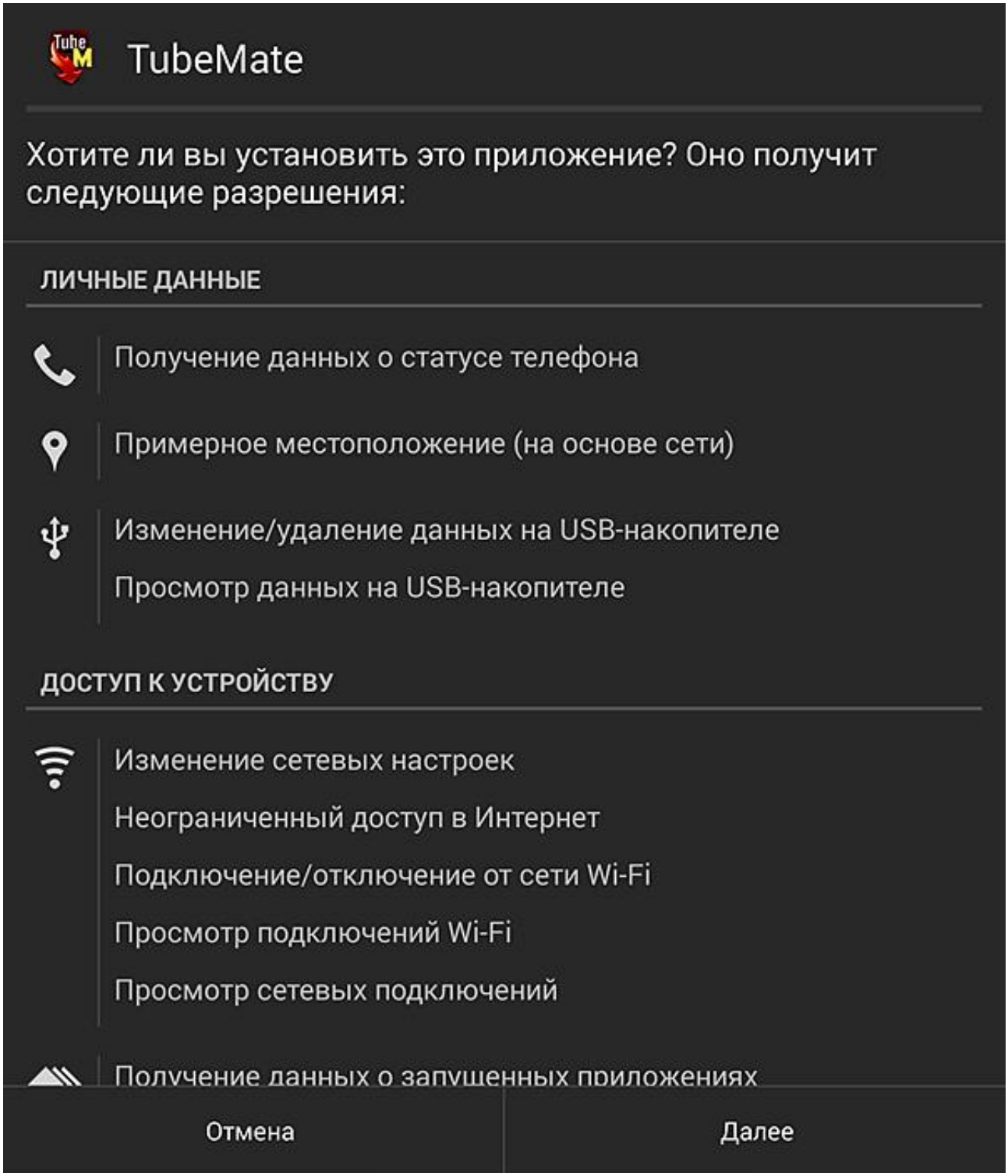


22. Получить QR-код программы

23. Запустить на мобильном устройстве MIT AI2 Companion App и просканировать QR-код приложения



24. Установить приложение на мобильное устройство.



24. После завершения установки, открыть приложение при помощи иконки на рабочем столе или в меню «Все приложения».

Глава 2. Практические приемы создания приложений

Объекты (компоненты) приложения могут реагировать на определенные события: нажатие на какую-либо клавишу, щелчок мышью по объекту, прикосновение к экрану или достижение края экрана и т.п.

После определенного события могут происходить какие либо действия.
Основным средством для обработки события являются так называемые блоки заголовков: Когда кнопка... Например Когда Кнопка Шелчок (событие) - проигрывается Звук (действие)

Некоторые события могут не зависеть от пользователя, к ним относятся:

- События таймера,
- События датчика, например, координаты GPS
- События на устройство, например входящие сообщения.
- События связанные с анимацией объектов, например достигнут край или происходит наложение объектов.
- Анимация события, такие как два объекта столкновения
- Веб-события, связанные с данными поступающими из сети

2.1 Кнопки

Кнопки являются наиболее часто используемым компонентом и используются для запуска различных действий.

События которые могут происходить с компонентом Кнопка по инициативе пользователя включают в себя, следующие

- Щелчок
- В фокусе
- Потерян Фокус
- Долгое нажатие
- Провести вниз
- Провести вверх

Примеры приложений

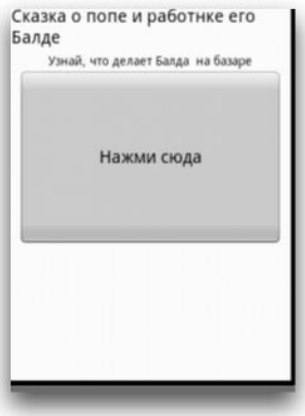
Используемые компоненты

- Табличное расположение
- Изображения
- Звуки
- Надпись

Пример 2.1.1 Приложение “Загадка”

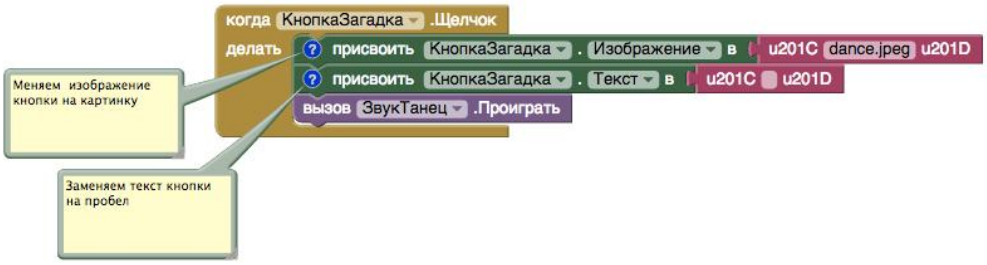
Описание. Приложения в котором, при нажатии на кнопку меняется изображение на ней.

Стартовый экран



Экран после нажатия кнопки

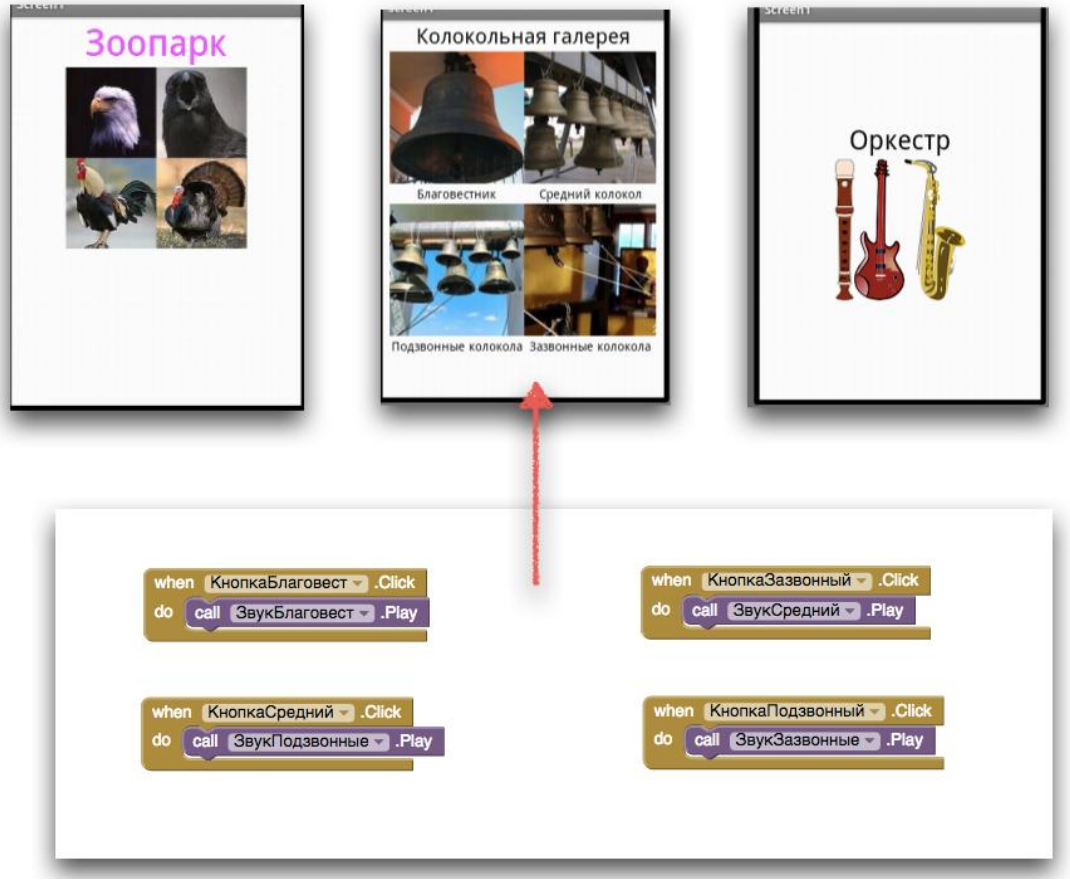
Блоки



Пример 2.1.2 Приложение “SoundBoard”

Описание: Приложение, в котором, при нажатии на соответствующие изображения, проигрываются соответствующие звуки.

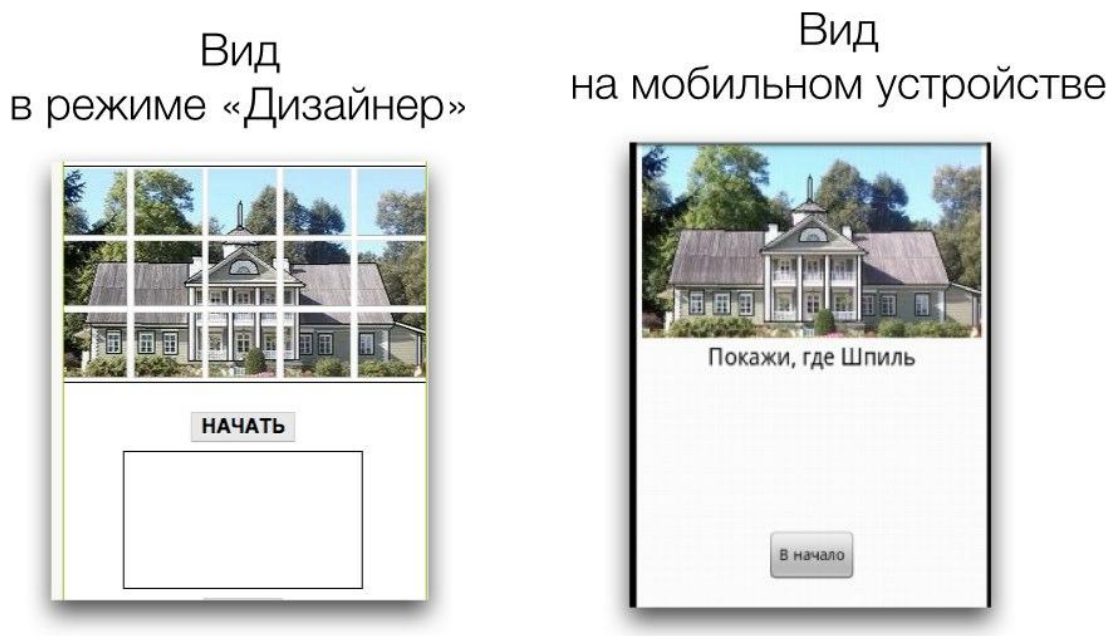
Примеры приложений “Зоопарк”, “Оркестр”, “Колокольная галерея”



Пример 2.1.3. Приложение “Отгадай-ка”

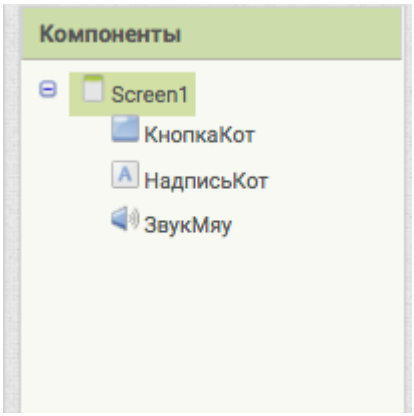
Описание: Тестовое приложение в котором, при нажатии на часть изображения выдается сообщение, соответствующее данной части изображения.

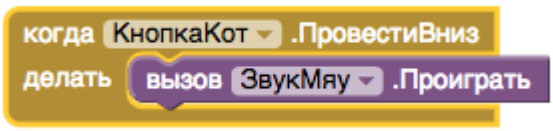
Одним из интересных приемов работы с изображением служит скрытое разделение изображения на части. Изображение, части которого необходимо выделить разрезается на нужное количество частей с помощью http://imagesplitter.net/ Полное изображение, создается из кнопок, каждая их которых содержит отдельный элемент. На основе таких приемов, можно строить тестовые задания с использованием изображений и кнопок.



Пример 2.1.4 Приложение “Виртуальный кот”

Описание: Приложение, в котором кот на экране издает звук когда его погладят.





2.2 Приложения с несколькими экранами

Создание приложения с несколькими экранами аналогично созданию нескольких отдельных приложений.

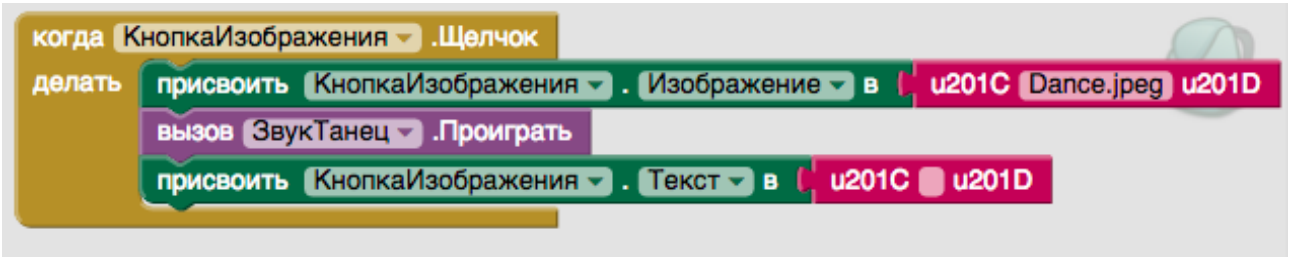
- Количество создаваемых в приложении экранов определяется разработчиком, но не может быть более 10. По умолчанию в момент создания нового проекта в нем всего один экран.
- Компоненты каждого экрана создаются в режиме “Дизайнер” для этого экрана. В режиме “Блоки” отображаются блоки, только для компонент текущего экрана.
- При создании приложений с несколькими идентичными экранами, используйте функцию копирования блоков между экранами.
- Навигация (переход) между экранами организуется с помощью кнопок или с помощью действий.
- Каждый экран, закрывается в случае перехода на другой или возвращении на тот, с которого он был открыт.
- Экраны могут обмениваться информацией путем принятия и возвращения значений, когда они открываются и закрываются. Экраны могут использовать для хранения и получения с других экранов данных только с использованием TinyDB, к примеру счет игры, количество набранных баллов в тесте.

Копирование блоков между экранами

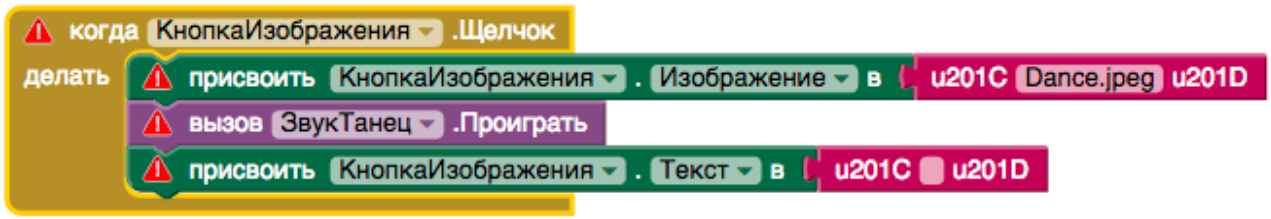
Некоторые приложения могут содержать идентичные экраны, с таким же набором компонент и аналогичными действиями.
В этом случае есть возможность копирования блоков программы с одного экрана на другой.
Алгоритм копирования включает следующие шаги.

1. Перейти в режим Блоки
2. На экране, перенести нужный набор блоков в “Рюкзак”

3. Перейти на другой экран.
4. Вновь нажать на Рюкзак и вынести нужные блоки на экран



5. Скопированные на другой экран блоки отображаются со значками предупреждения



6. После настройки компонент и привязки их к конкретному экрану знаки предупреждения будут удалены.

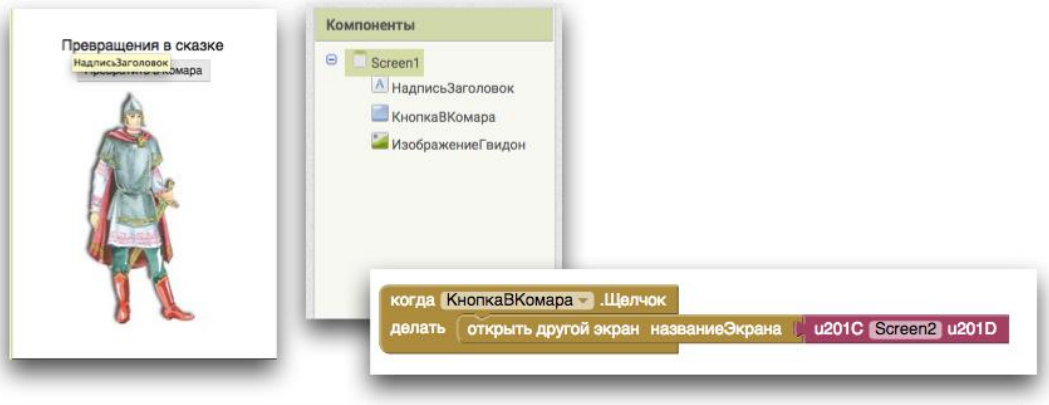
Пример 2.2.1 Приложение “Сказочные превращения”

Описание. Превращение одного изображения в другое “Муха” в “Слона”, “Гвидон” в “Комара”.

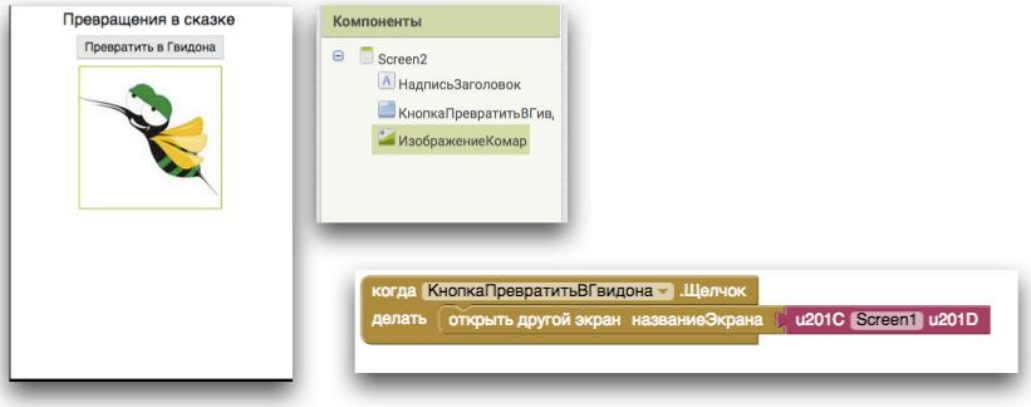
Компоненты приложения:

- Экраны: Screen1 и Screen 2
- Надпись
- Кнопка
- Изображение

Экран 1



Экран 2



2.3 Обмен данными между экранами

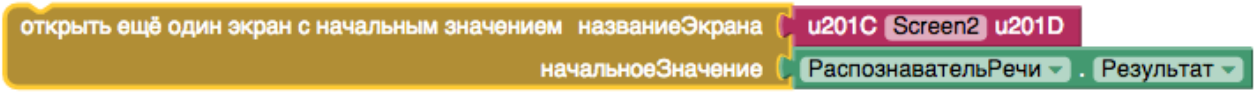
Способ 1. Использование компонента **TinyDB**

Компонент TinyDB используется внутри приложения для передачи данных между экранами, В этом их отличие от глобальных переменных, которые сохраняются в пределах одного экрана, пока приложение работает.

- Компонент TinyDB очень полезен, поскольку позволяет сохранить данные приложения на Android устройстве. Обычно такие небольшие данные используются для сохранения настроек приложения. Что вам нужно знать об этом компоненте: компонент невидимый, он никак не отображается; у вас только одно хранилище данных. Если вы поместите несколько компонентов TinyDB, то получите доступ к одному и тому же набору данных.
- С помощью TinyDB вы не можете получить доступ к данным другого приложения.
- После того как данные сохранены в TinyDB они останутся там, пока TinyDB не будет очищено.

Способ 2. Использование функции открытия экрана с начальным значением

При передаче данных между экранами можно использовать обработчик события для экрана
На экране с которого нужно передать данные, необходимо в этом блоке указать нужные переменные или значения:



На другом экране можно получить их с помощью функции получить начальное значение, к примеру при инициализации экрана:

Пример 2.3.1 Приложение “Сказочные перемещения”

Описание. Приложение: котором происходит перемещение объекта с одного экрана на другой. Переместить Гвидона с острова в палаты (возможны примеры “Фокусник” зайца из шляпы, стакан воды перенести и другие)

Компоненты приложения:

- Экраны: Screen1 и Screen2
- Надписи
- Холст
- ИзображениеСпрайт
- Кнопки
- TinyDB

Пример 2.3.2 Приложение “Хамелеон”

Описание.
Приложение состоящее из двух экранов, в котором при щелчке по кнопке экран закрашивается выбранным цветом, и при переходе на второй экран приложение сохраняет цвет первого экрана, и закрашивает второй экран этим же цветом.

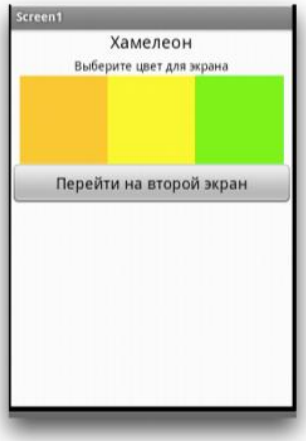
Компоненты приложения:

- Экраны: Screen1 и Screen2
- Надписи
- Кнопки
- TinyDB
- Цвет

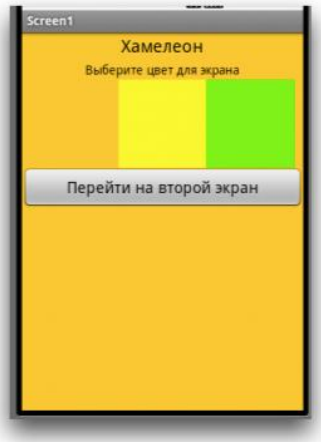
Приложение «Хамелеон»

Интерфейс

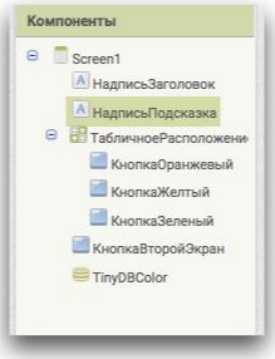
Screen 1.
До выбора цвета



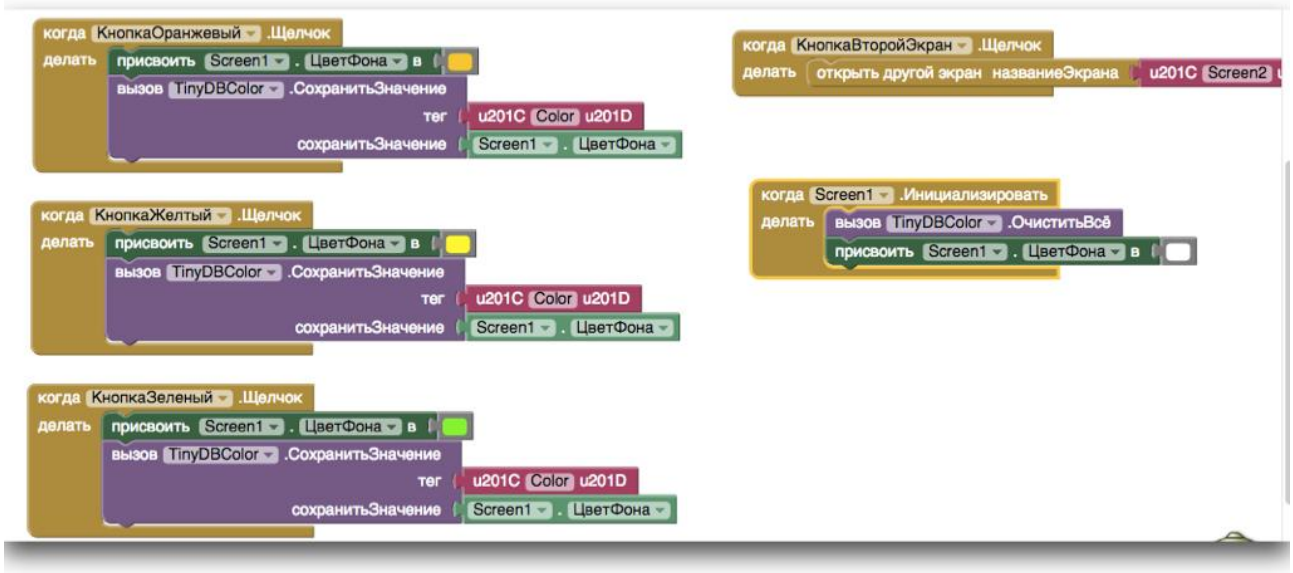
Screen 1.
После выбора цвета

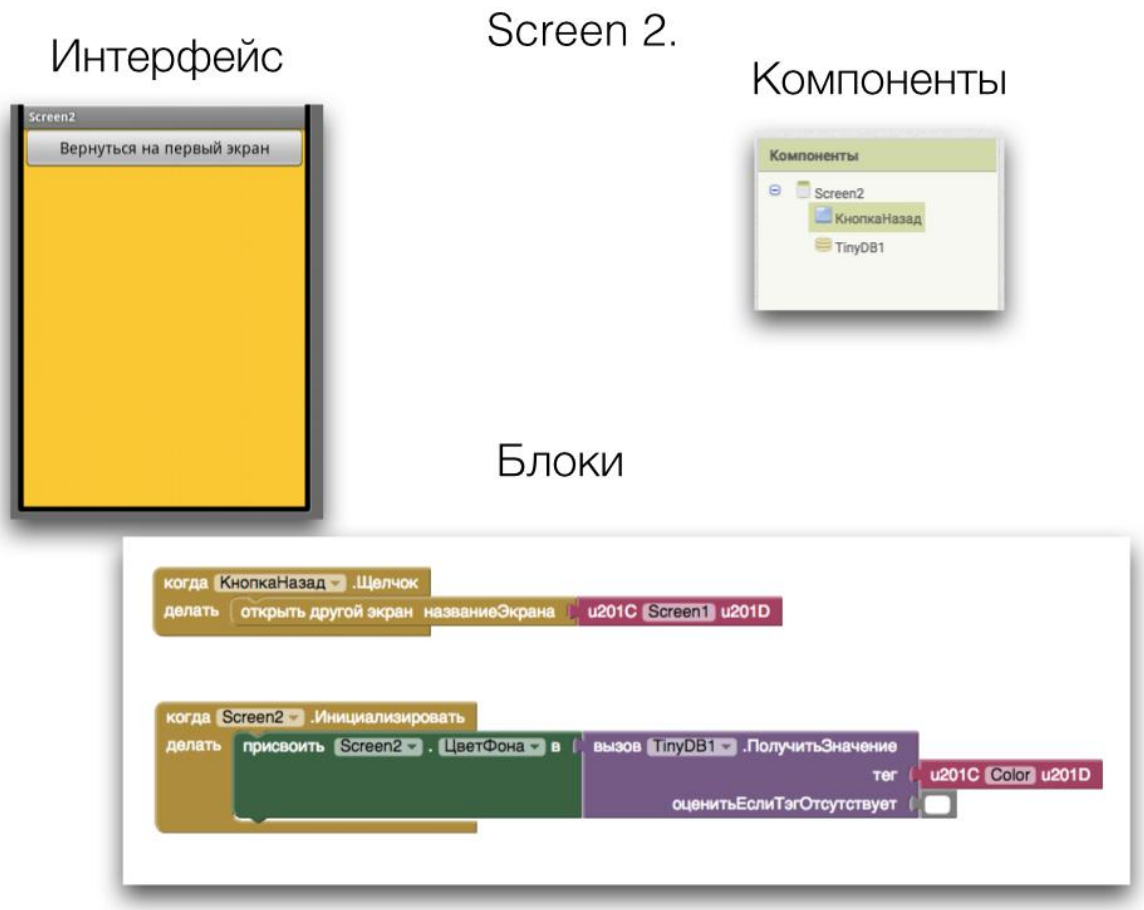


Компоненты



Блоки


когда КнопкаОранжевый .Щелчок
делать присвоить Screen1 . ЦветФона в
вызов TinyDBColor .СохранитьЗначение
тег u201C Color u201D
сохранитьЗначение Screen1 . ЦветФона
когда КнопкаЖелтый .Щелчок
делать присвоить Screen1 . ЦветФона в
вызов TinyDBColor .СохранитьЗначение
тег u201C Color u201D
сохранитьЗначение Screen1 . ЦветФона
когда КнопкаЗеленый .Щелчок
делать присвоить Screen1 . ЦветФона в
вызов TinyDBColor .СохранитьЗначение
тег u201C Color u201D
сохранитьЗначение Screen1 . ЦветФона
когда КнопкаВторойЭкран .Щелчок
делать открыть другой экран названиеЭкрана u201C Screen2
когда Screen1 .Инициализировать
делать вызов TinyDBColor .ОчиститьВсё
присвоить Screen1 . ЦветФона в

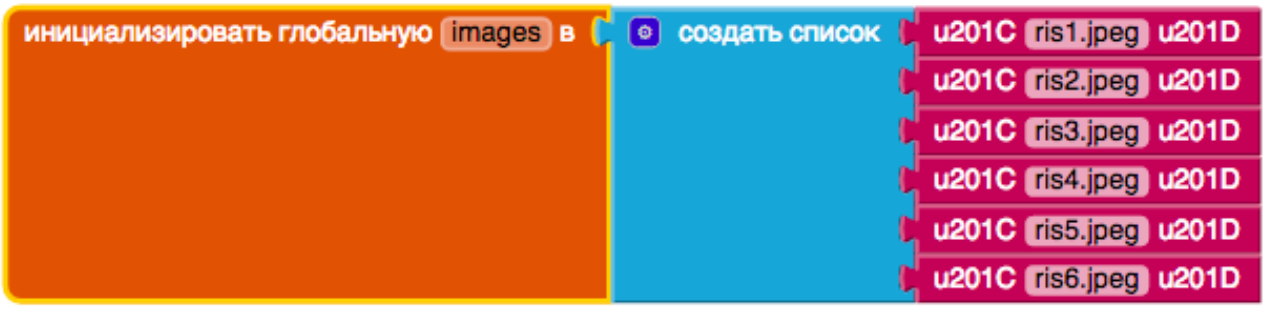


2.4 Списки

Использование списков в приложении позволяет создавать нестандартные цвета для компонент и их свойств, создавать записные книжки для хранения данных: различных фильмов, произведений, телефонов друзе и многое другое.
В некоторых приложениях необходимо вводить переменные.

Переменные

- Переменные - это область памяти, в которой можно хранить какие - либо значения.
- Переменная должна иметь свое уникальное имя, в приложении на одном экране не может быть двух переменных с одинаковыми именами .
- Переменная в приложении определяется с помощью блока Инициализировать.



- Информация хранящаяся в переменной может изменяться.

Локальная переменная - это переменная, которая инициализирована внутри функции или она может быть аргументом, переданным в функцию или процедуру. Доступ к локальной переменной возможен только к этим переменным в конкретной функции или процедуре, в которой они были заданы или передан в качестве аргумента.

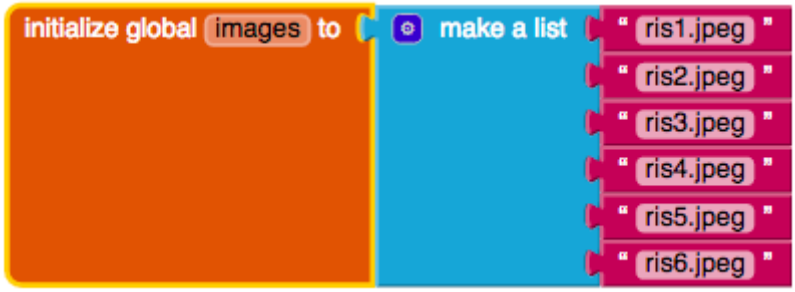
Глобальная переменная - это переменная, которая может быть доступна в нескольких областях программы внутри одного экрана. Внутри программы можно получить текущее значение переменной или задать новое значение.

Списки (в английской версии List или одномерные массивы) одни из самых интересных компонент приложения.

Списки/ массивы представляют собой определенный тип структуры данных, которые используются во всех языках программирования.
Массивы можно использовать для создания и управления различными наборами значений / компонент.

Индекс

Положение элемента в списке называется его индексом. В Mit App Inventor, первый элемент в списке имеет индекс 1.
При создании приложений можно ссылаться на конкретный элемент внутри списка, если знаем, какой индекс имеет элемент в данном списке.



В приложении кроме набора стандартных цветов, есть возможность создания собственного цвета. Такая функция может быть реализована с помощью списка.

RGB (аббревиатура английских слов Red — красный, Green- зелёный, Blue синий) — аддитивная цветовая модель, как правило, описывающая способ синтеза цвета для цветовоспроизведения. При задании цвета в App Inventor - может быть заданное четвертое значение является дополнительным и представляет значение альфа или как насыщенный цвет. По умолчанию альфа-значение по 100.

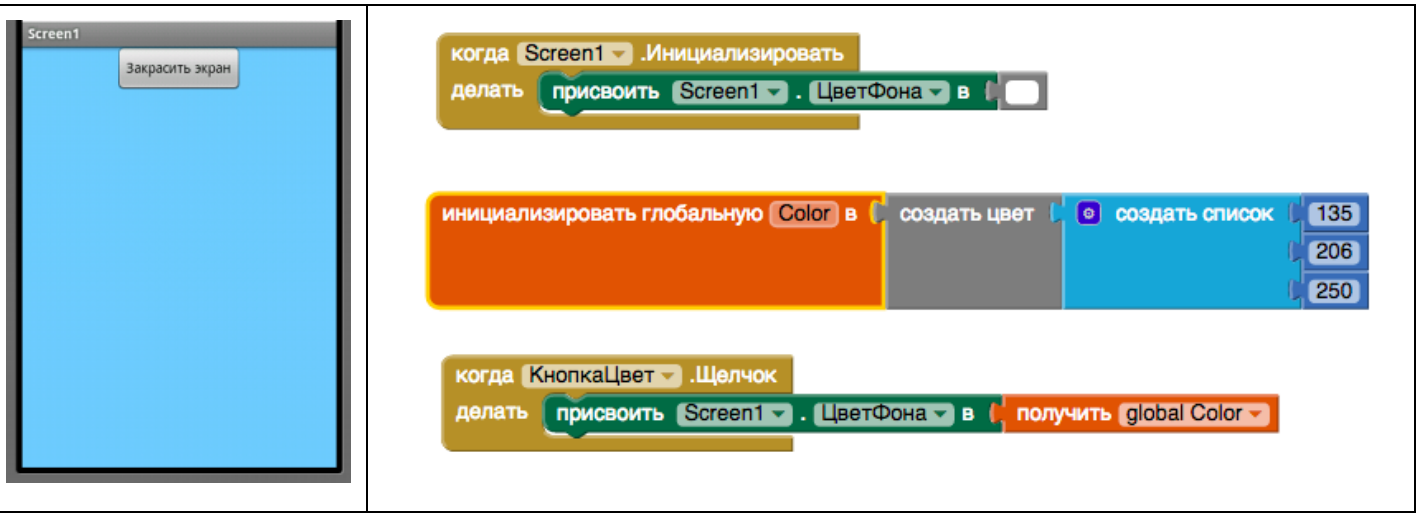
MediumSlateBlue	123 104 238
LightSlateBlue	132 112 255
MediumBlue	0 0 205
RoyalBlue	65 105 225
Blue	0 0 255
DodgerBlue	30 144 255
DeepSkyBlue	0 191 255
SkyBlue	135 206 235
LightSkyBlue	135 206 250
SteelBlue	70 130 180

Пример 2.4.1 Создание собственного цвета

Описание. Приложение, в котором экран закрашен цветом, созданным самим пользователем.

Компоненты приложения:

- Экран: Screen1
- Кнопка

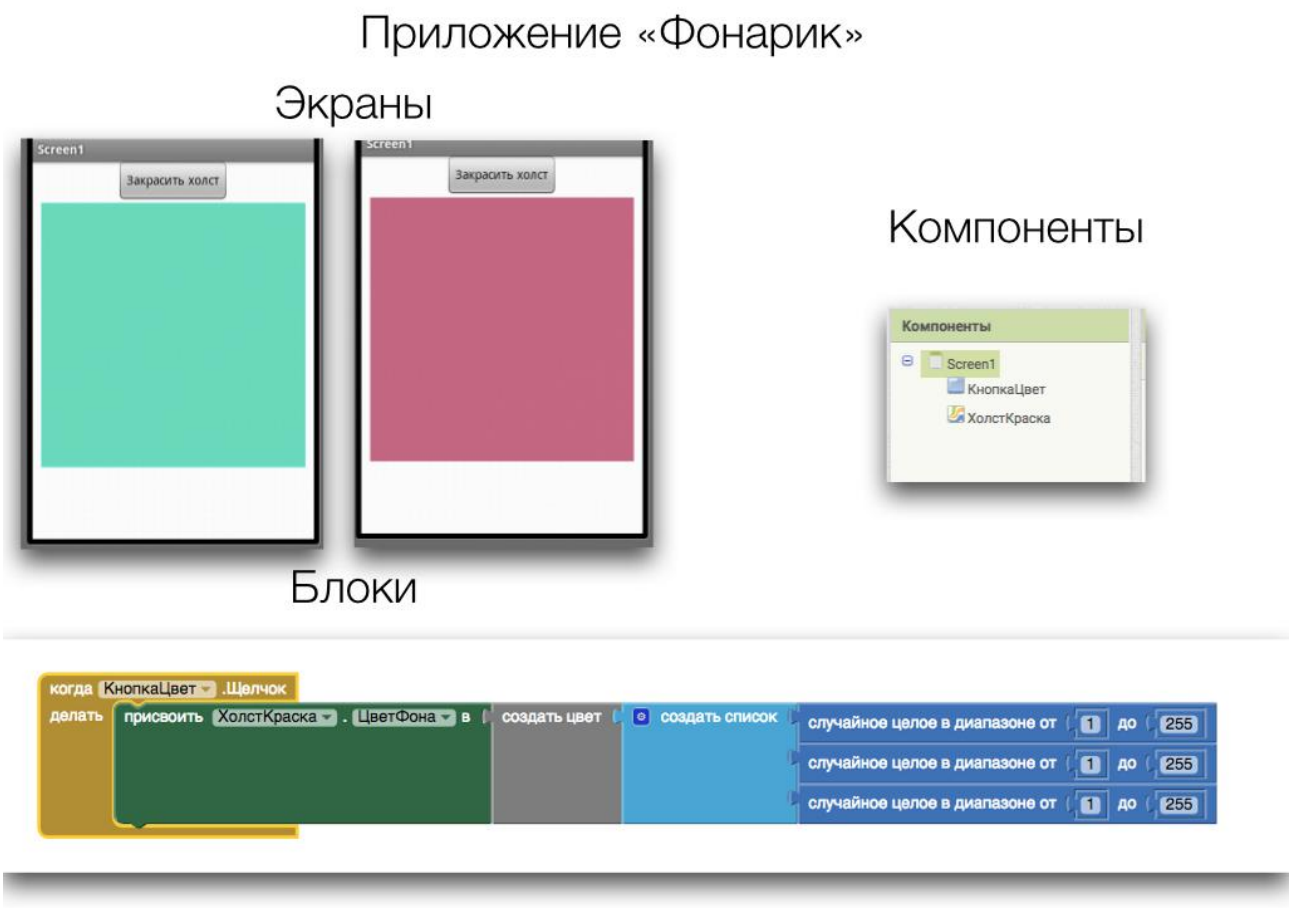


Пример 2.4.2 Приложение “Фонарик”

Описание. Приложение, в котором холст закрашивается случайно сгенерированными цветами.

Компоненты приложения:

- Холст
- Кнопка

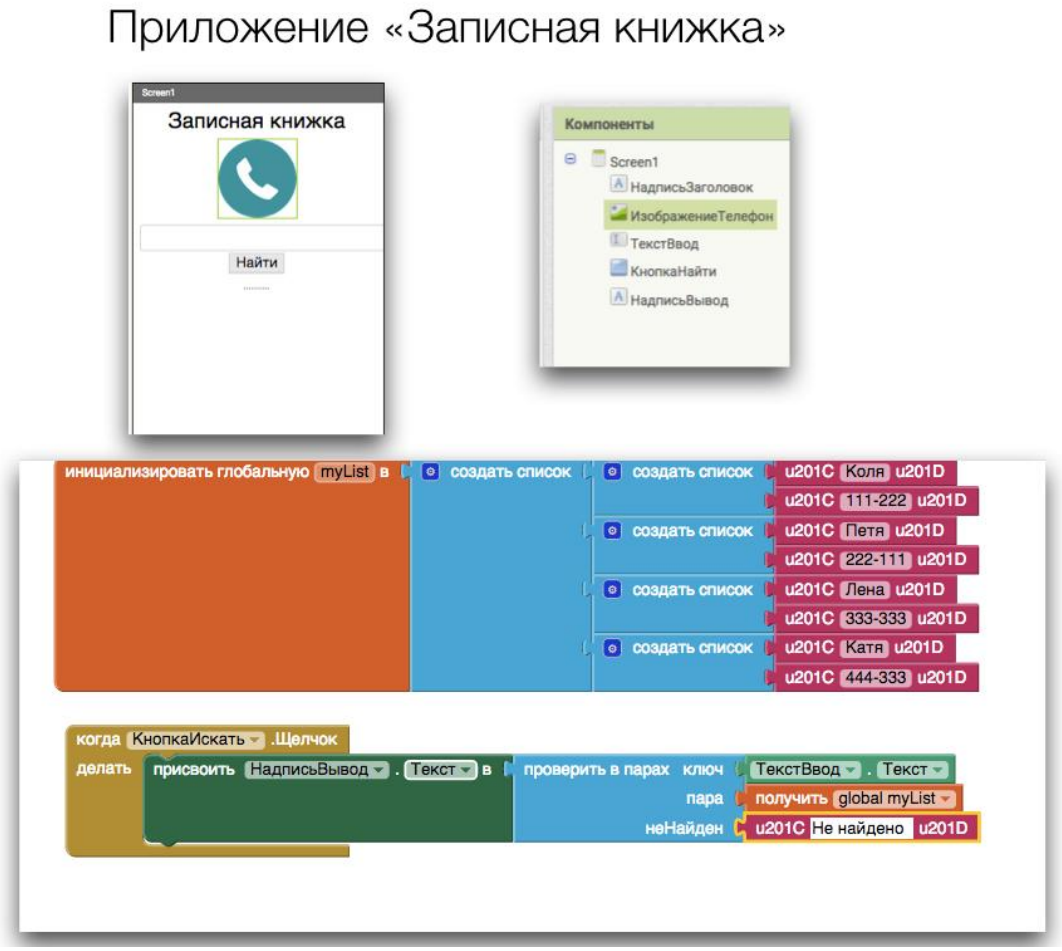


Пример 2.4.3 Приложение “Записная книжка”

Описание. Приложение в котором при вводе имени друга выводится его телефон (или любимый фильм, или день рождения)

Компоненты приложения:

- Надпись
- Изображение
- Текст
- Кнопка



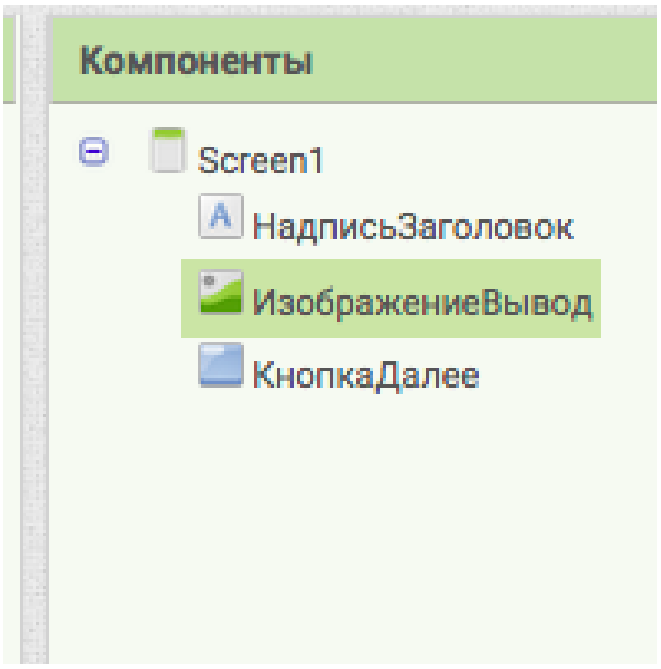
Пример 2.4.4 Приложение “Слайд-шоу”

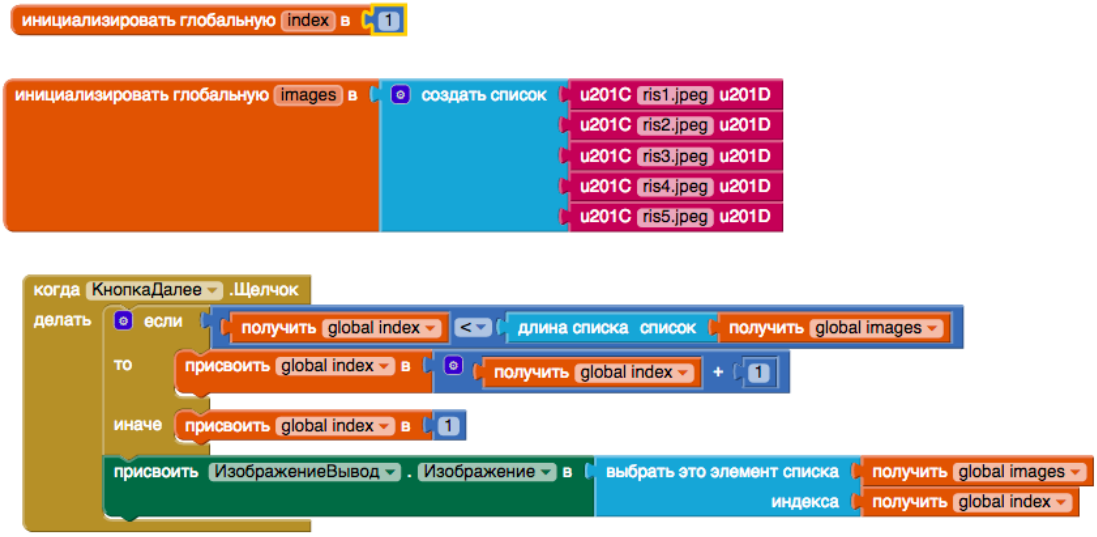
Описание. Приложение , отображающее слайд шоу из изображение.

Компоненты приложения:

- Надпись
- Изображение
- Кнопка


Screen1
Самые загадочные здания
Далее


Компоненты
Screen1
НадписьЗаголовок
ИзображениеВывод
КнопкаДалее


инициализировать глобальную index в 1
инициализировать глобальную images в создать список
u201C ris1.jpeg u201D
u201C ris2.jpeg u201D
u201C ris3.jpeg u201D
u201C ris4.jpeg u201D
u201C ris5.jpeg u201D
когда КнопкаДалее .Щелчок
делать если получить global index < длина списка список получить global images
то присвоить global index в получить global index + 1
иначе присвоить global index в 1
присвоить ИзображениеВывод . Изображение в выбрать это элемент списка получить global images
индекса получить global index

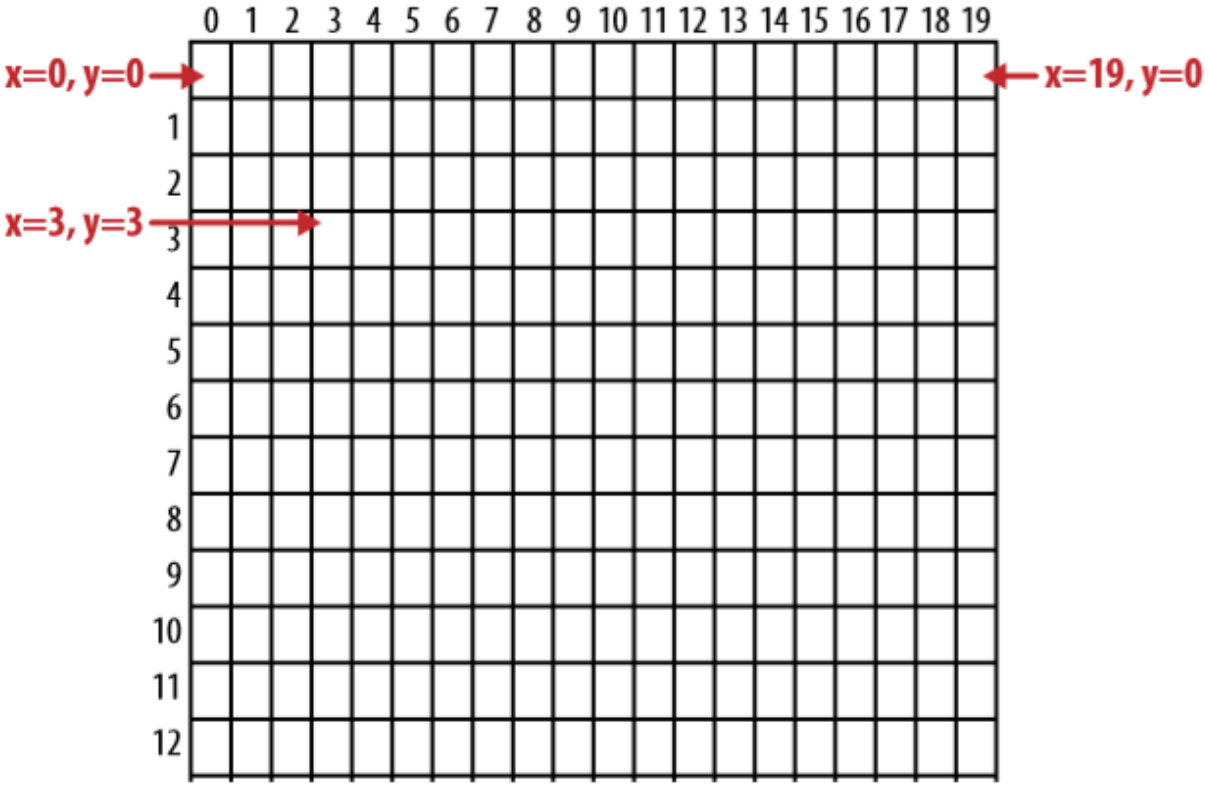
2.5 Рисование

Основные тезисы:

Холст

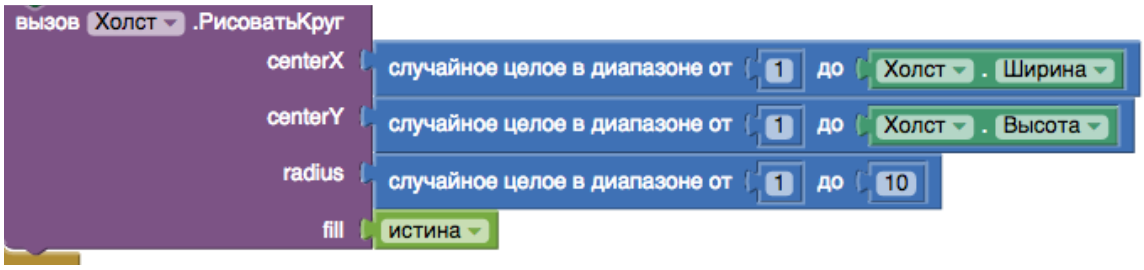
- Компонент Холст является дополнительной панелью приложения.
- Холст используется для рисования на нем объектов. размещения. текста и анимации изображений - спрайтов.

- Отображение холста во весь экран требует установки параметров “Наполнить родительский” при задании свойств Высоты и Ширины Холста.
- Размещение дополнительных компонент на экране, кроме холста, требует установки фиксированных значений его ширины и высоты.
- Местоположение объекта на холсте определяется значениями X, Y значение по отношению к левой верхней углу холста. X представляет горизонтальное положение объекта, 0 является левой границей и X увеличивается, когда объект перемещается вправо. У - вертикальная положение объекта, 0 верхняя граница и Y увеличивается, когда объект перемещается вниз.



- Рисование круга/окружности на холсте требует указания параметров координаты X и Y центра круга, а также радиуса в пискелах и параметра заливки (ложь/истина).
- Рисование линий на холсте требует установки параметров X и Y начала и конца линии.

- При использовании случайных значений координат и в случае когда ширина холста определяется параметром “Наполнить родительский! , рекомендуется использовать функции определения ширины и высоту холста для устройства, как показано на примере:

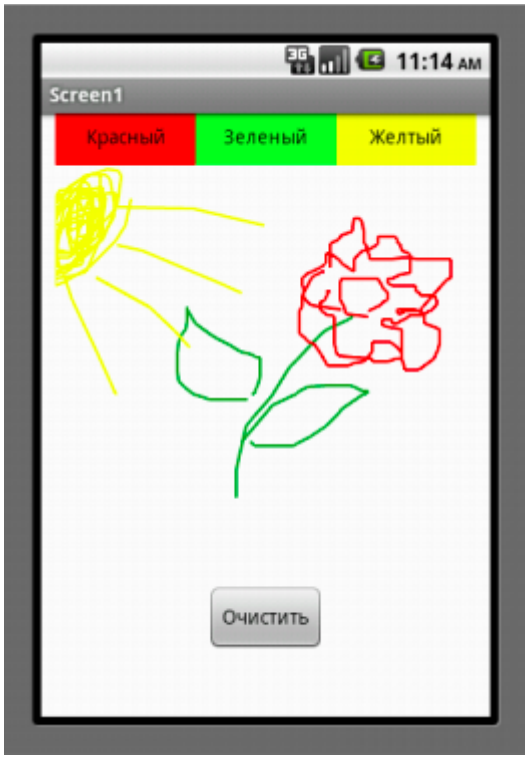


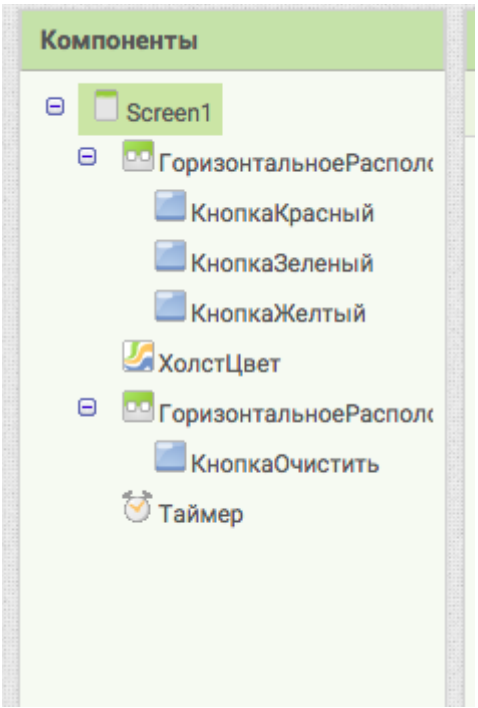
Пример 2.5.1 Приложение “Рисование”

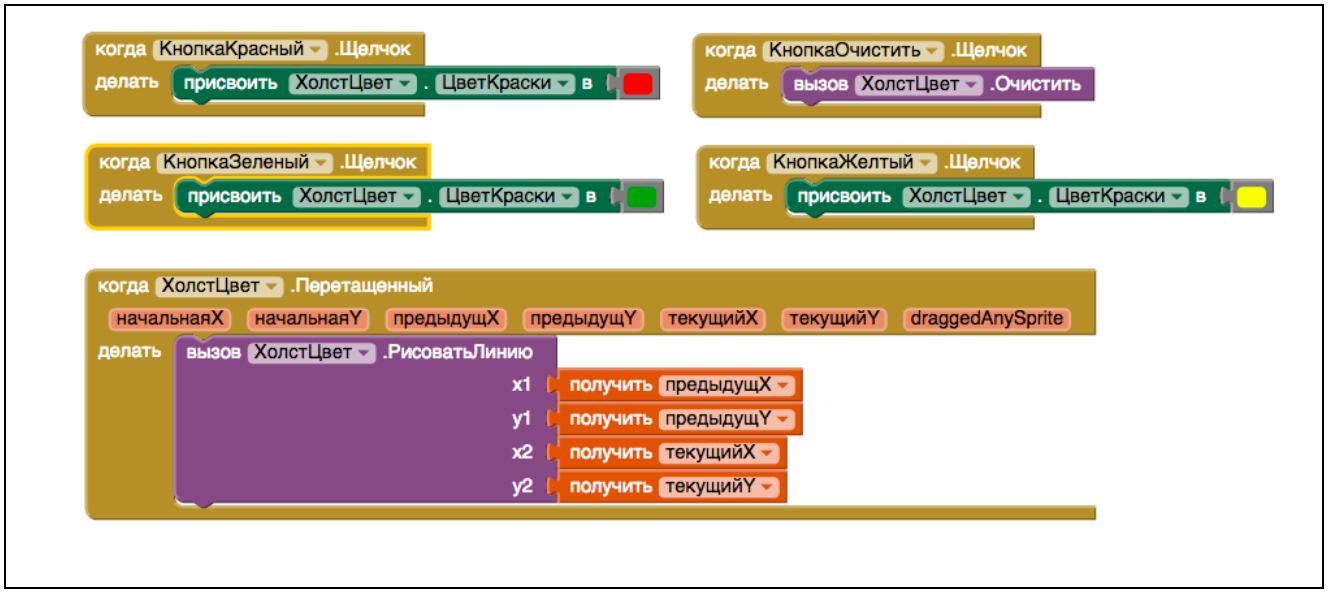
Описание. Приложение, позволяющее пользователю рисовать на экране.

Компоненты

- *Кнопка*
- *Холст*

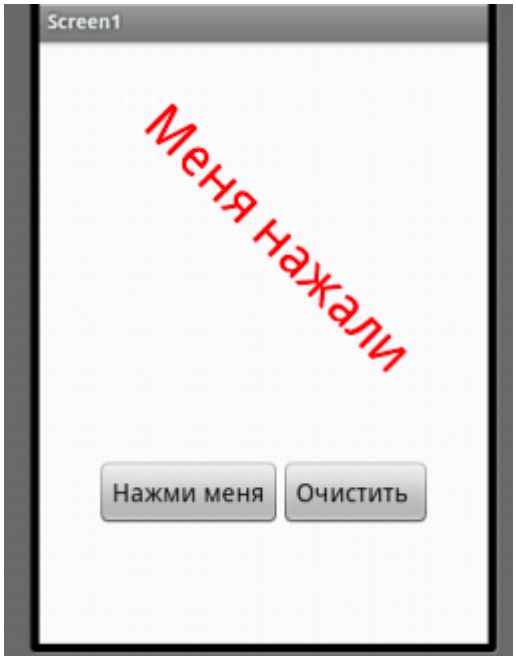


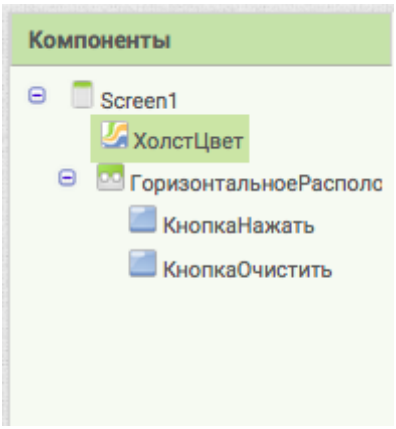


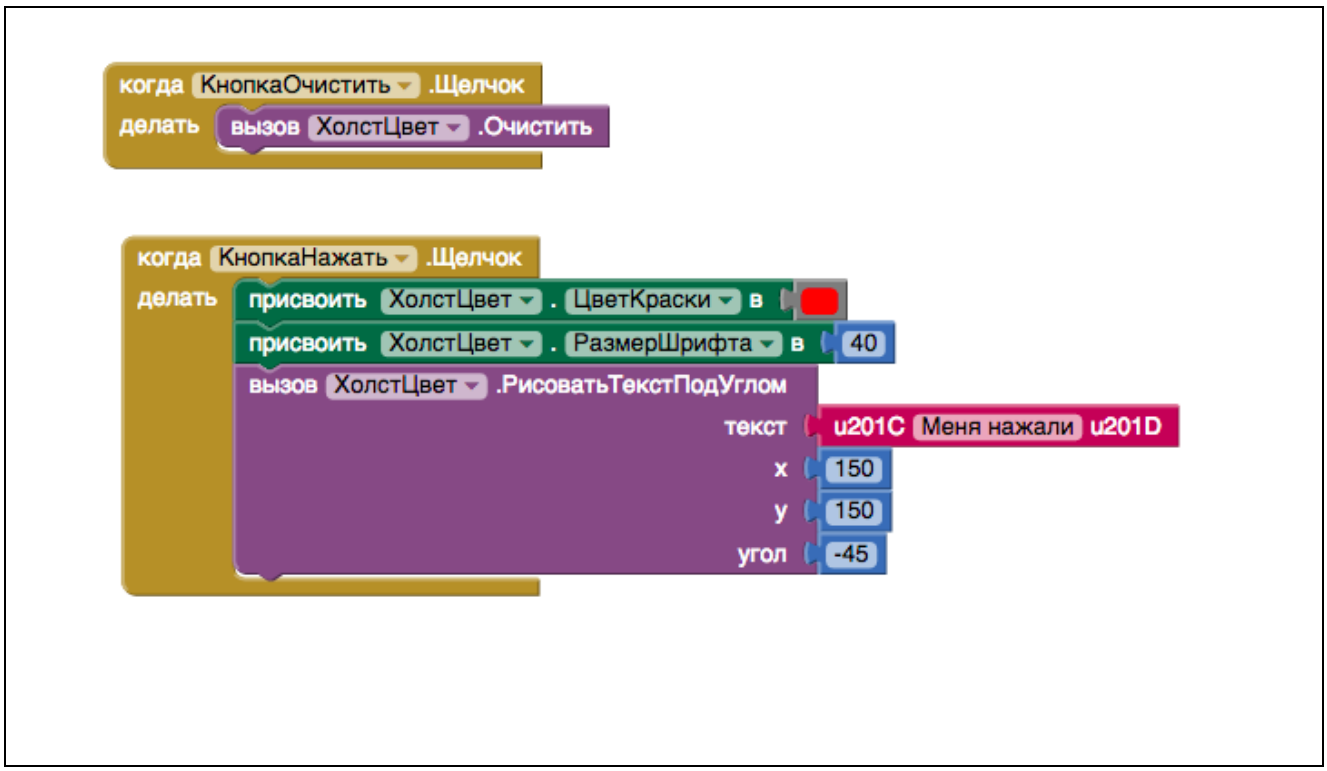


Пример 2.5.2 . Приложение “Пишем на холсте”

Описание. Приложение, в котором при нажатии кнопке на холсте под углом отображается текст “Меня нажали”

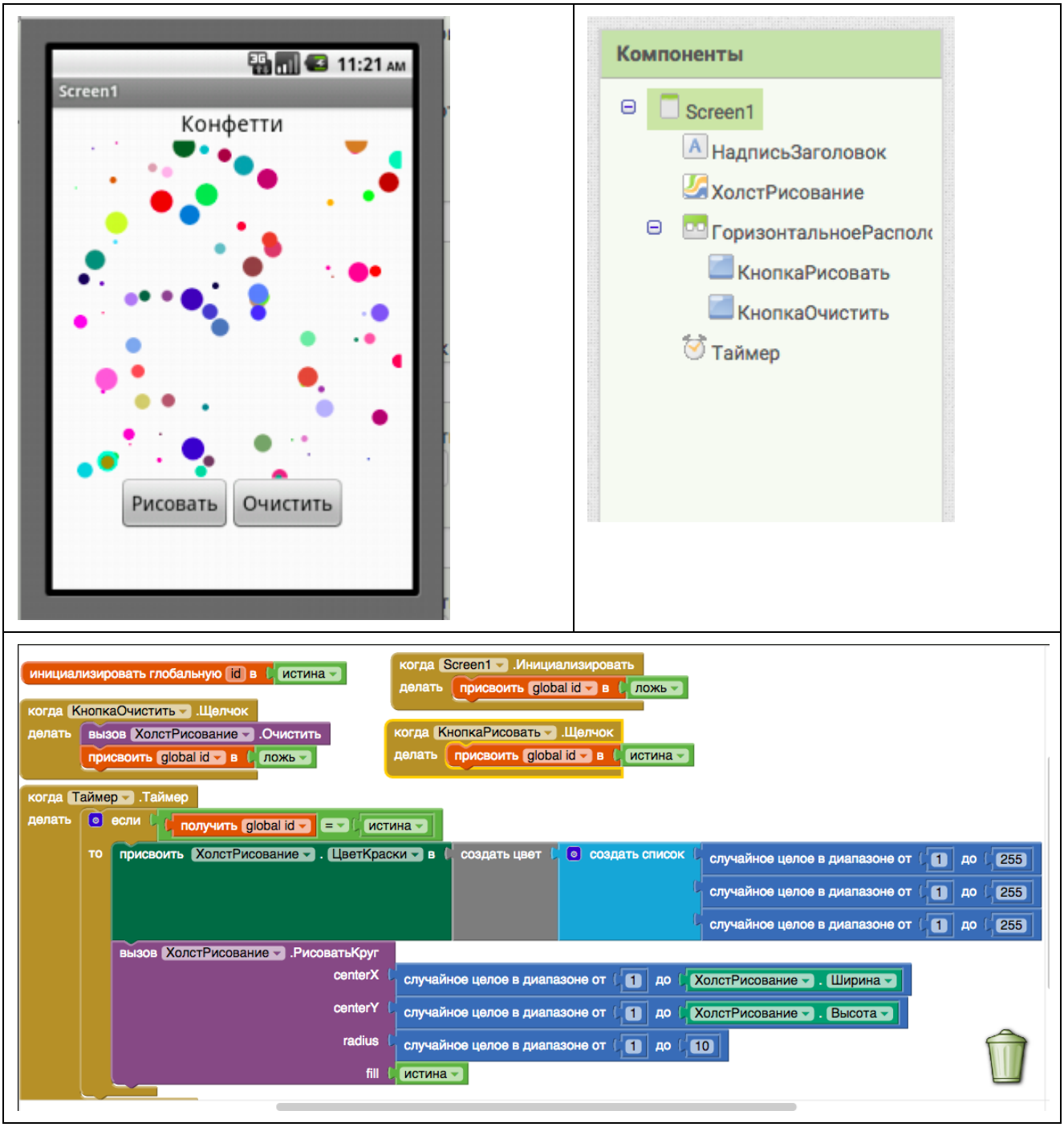





когда КнопкаОчистить .Щелчок
делать вызов ХолстЦвет .Очистить
когда КнопкаНажать .Щелчок
делать присвоить ХолстЦвет . ЦветКраски в
присвоить ХолстЦвет . РазмерШрифта в 40
вызов ХолстЦвет .РисоватьТекстПодУглом
текст u201C Меня нажали u201D
x 150
y 150
угол -45

Пример 2.5.3 . Приложение“ Конфетти”

Описание. Приложение, в котором при нажатии кнопки холст случайным образом закрашивается точками различного диаметра и цвета.

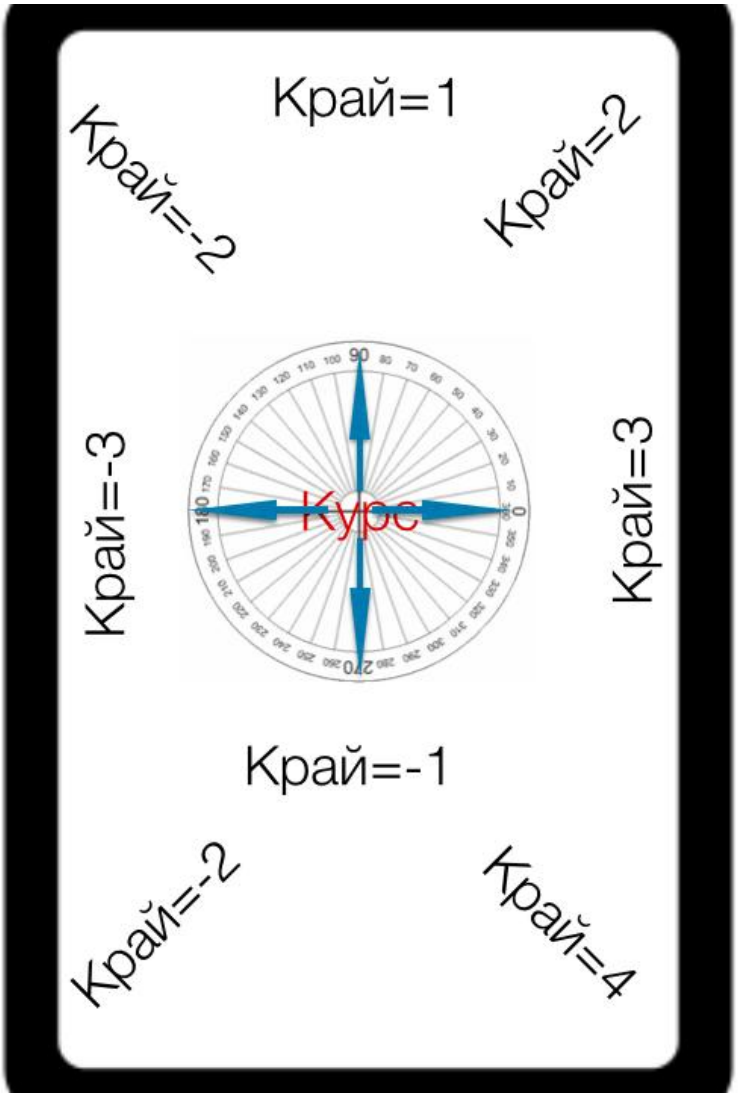


2.6 Анимация

Основные тезисы

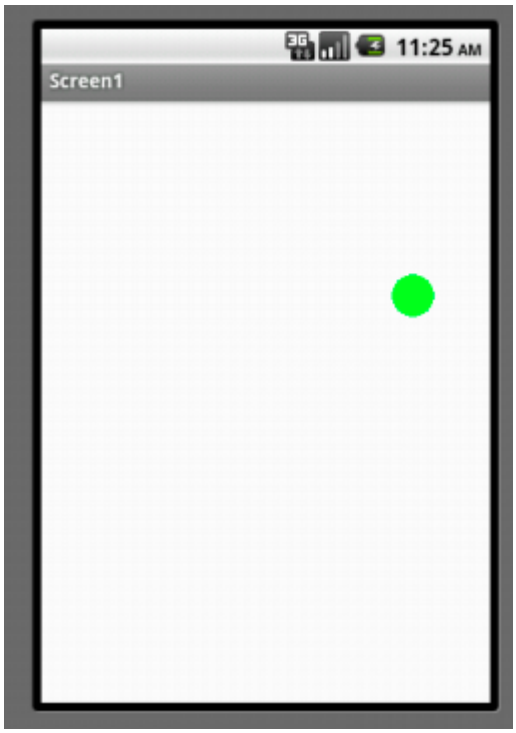
Холст является обязательным компонентом приложений с анимированными объектами.

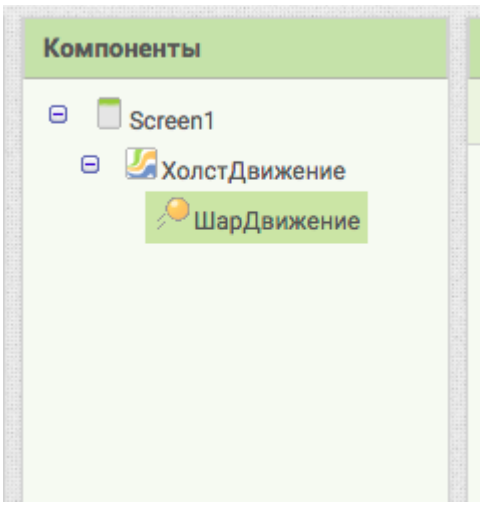
- Для анимации предлагается два компонента - “Шар” и “ИзображениеСпрайта”. Шар это круг, для которого можно установить такие свойства как радиус и цвет. ИзображениеСпрайта - прямоугольное изображение, в качестве источника которого, может быть установлено любое загруженное изображение.
- Оба этих объекта должны быть размещены на холсте, и могут реагировать на прикосновения, перемещения, взаимодействовать с другими спрайтами и пр.
- На холсте может находиться множество компонент данного типа.
- Существуют определенные правила задания угла движения спрайта и определения края холста. Числовые значения края устанавливаются на основании следующей схемы:

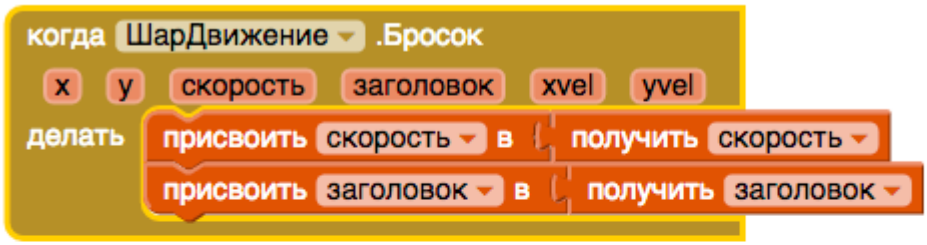


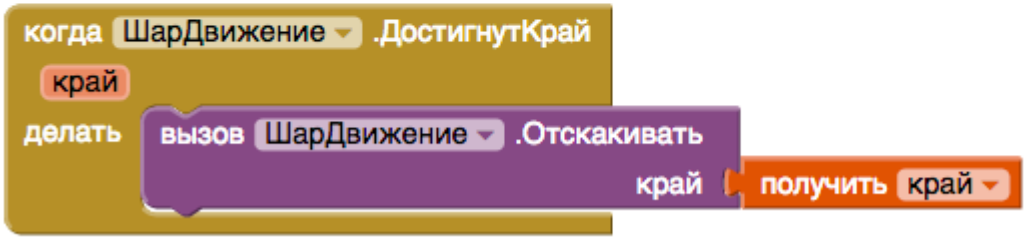
Пример 2.6.1 Приложение “Игра в мяч”

Описание. Приложение в котором мяч двигается по экрану и при достижении края отскакивает от него и движется в обратную сторону



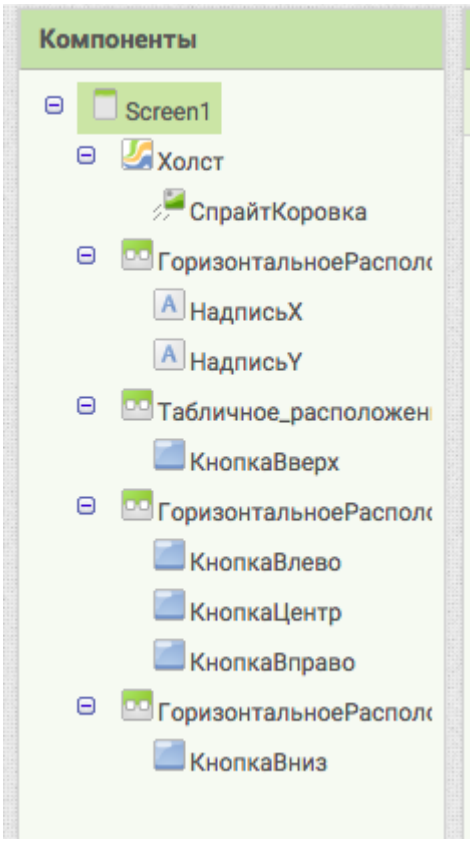


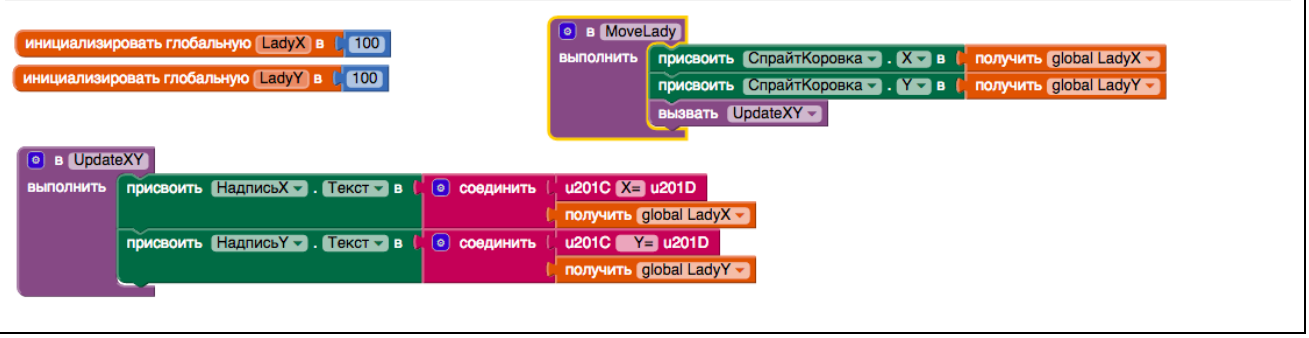


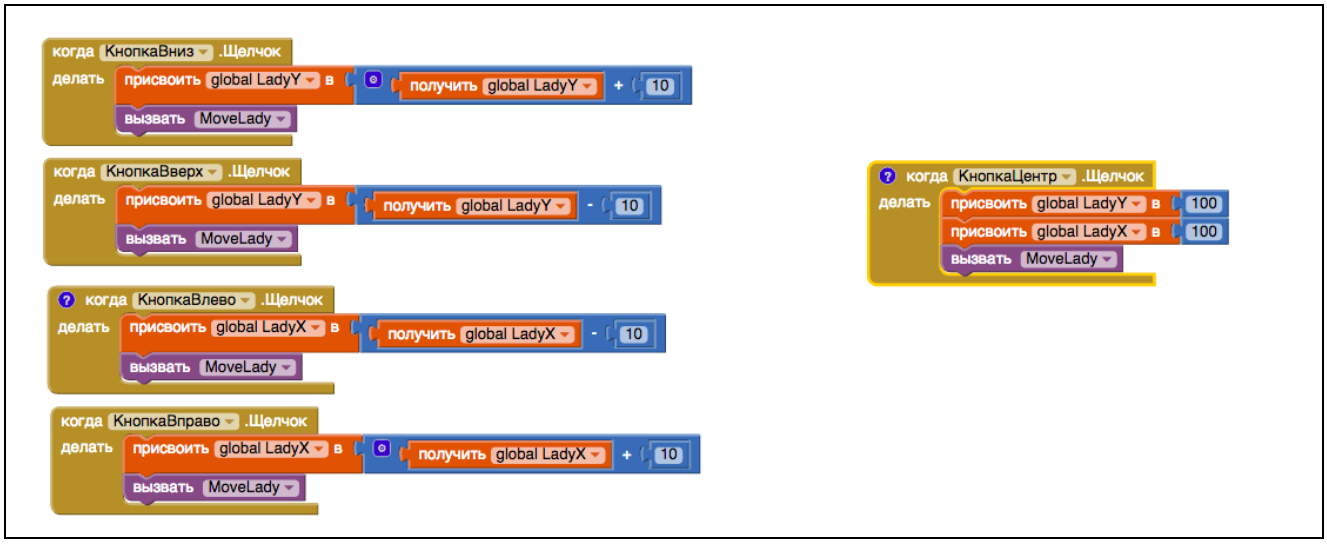


Пример 2.6.2. Приложение" Управляем движением объекта"

Описание, В приложении Изображение Спрайта на экране управляется движением кнопок







2.7 Медиа

Основные тезисы

Медиа компоненты позволяют использовать встроенные в мобильные устройства фото и видео ка меры, динамики, микрофоны и пр.

Медиакомпоненты включают в себя:

- **Проигрыватель-** предназначен для воспроизведения аудио/видеофайла или для вибрации телефона. Компонент Проигрыватель невидимый, его лучше всего использовать для воспроизведения аудиофайлов, для воспроизведения видео используется компонент VideoPlayer.

- **Звук** - применяется для воспроизведения коротких звуковых файлов вроде звуковых эффектов. Особенность этого компонента — свойство Минимальный Интервал, задающее минимальный интервал звучания, после которого звук будет повторяться.

- **ТекстВРечь** - компонент, позволяющий преобразовать Текст в речь

- **ЯндексПереводчик** - позволяет переводить тексты с одного языка на другой. Для работы с ними требуется подключение к сети Интернет. Язык перевода задается в формате en - английский, ru - русский, es- испанский , fr - французский.

- **Видеоплеер** используется для воспроизведения видеофайлов. Поддерживаются форматы: Windows Media Video (wmv), 3GPP (3gp), MPEG-4 (mp4). Максимальный размер файла — 1 Мбайт В отличие от компонентов Player и Sound, данный компонент отображается на экране

- **Камера** - используется для получения снимка со встроенной камеры устройства.

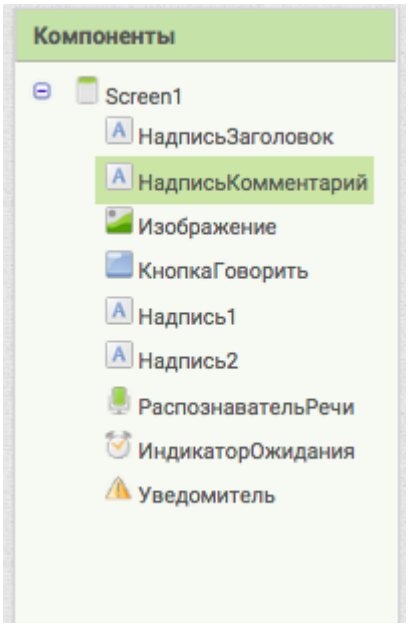
- **Распознаватель Речи** - компонент, позволяющий преобразовать речь в текст.

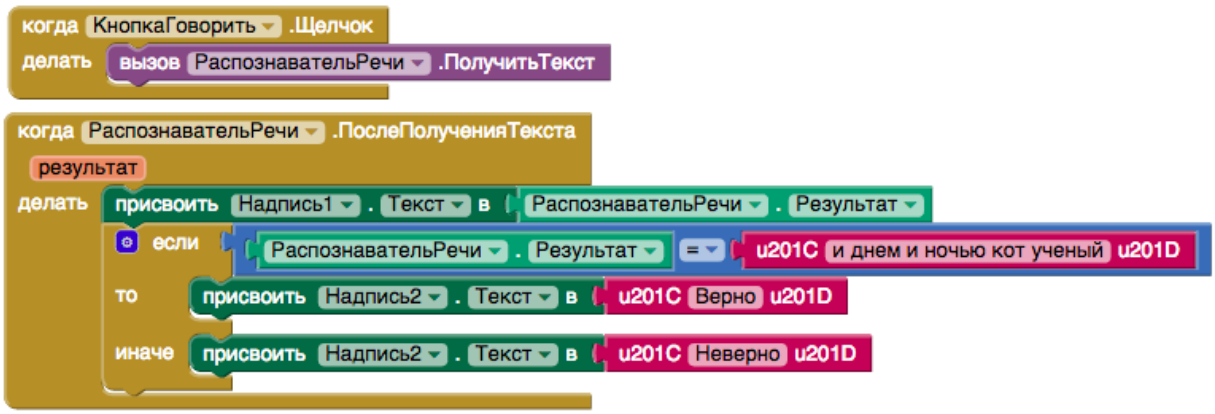
- **Выборщик изображений с**лужит для выбора изображения из вашей галереи изображений. Компонент похож на ListPicker, он тоже представляет собой кнопку, которая открывает окно выбора, только не элементов списка, а изображений.

- Диктофон - позволяет записать звук.

Пример 2.7.1 Приложение “Распознавание речи”

Описание приложения. Приложение проверяет строчку стихотворения, сверяя ее с оригиналом и выдает сообщение “Правильно” или “Неправильно”.

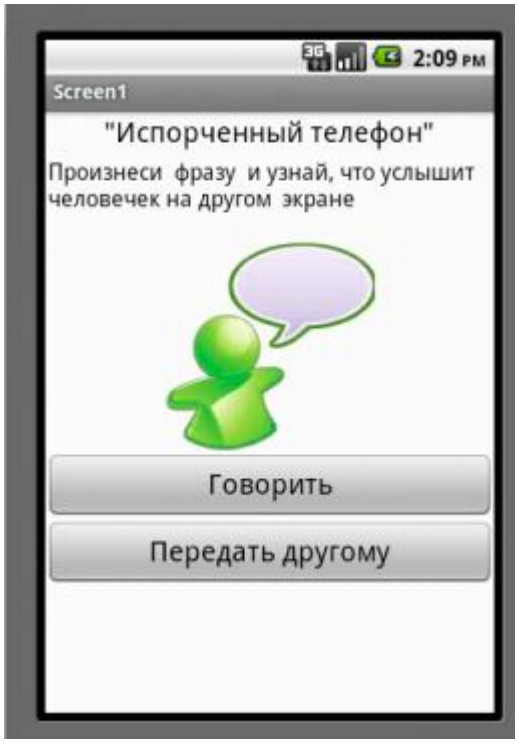


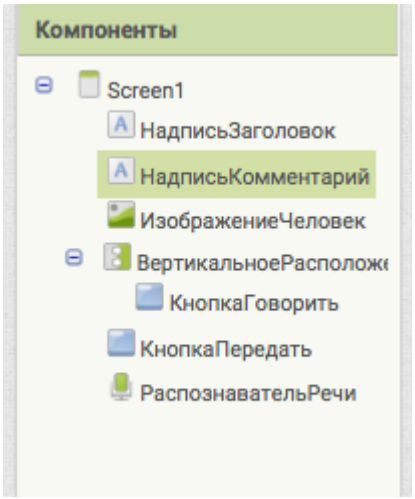


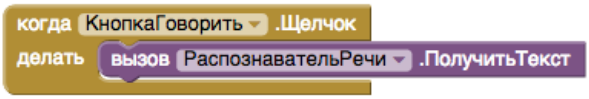
Пример 2.7.2 Приложение “Испорченный телефон”

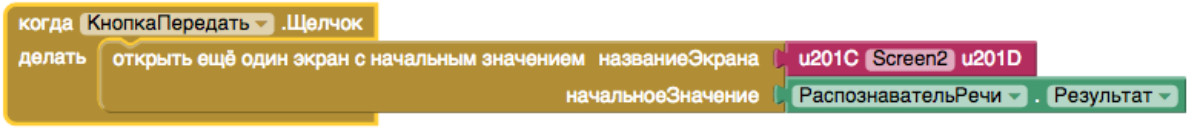
Описание: Приложение, которое будет передавать услышанный текст от одного собеседника (первый экран), второму собеседнику (второй экран) с использованием функции передачи значений между экранами.

Screen1

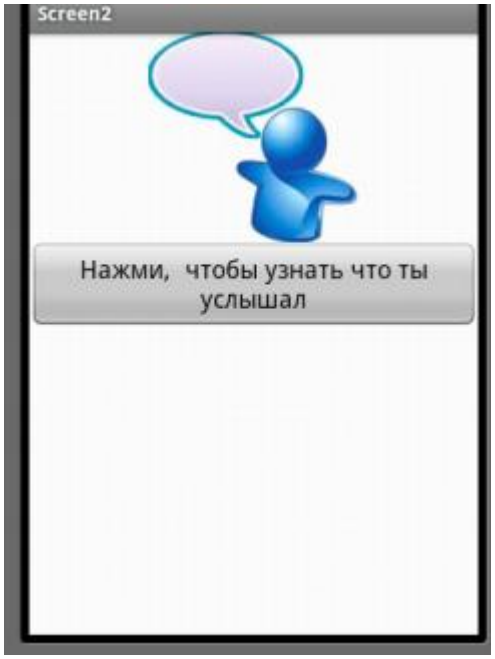

2:09 PM
Screen1
"Испорченный телефон"
Произнеси фразу и узнай, что услышит человечек на другом экране
Говорить
Передать другому

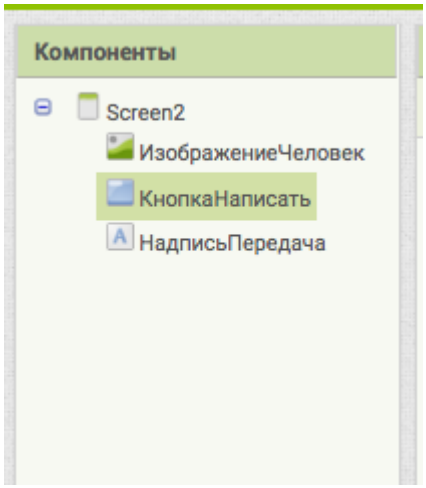

Компоненты
Screen1
НадписьЗаголовок
НадписьКомментарий
ИзображениеЧеловек
ВертикальноеРасположе
КнопкаГоворить
КнопкаПередать
РаспознавательРечи

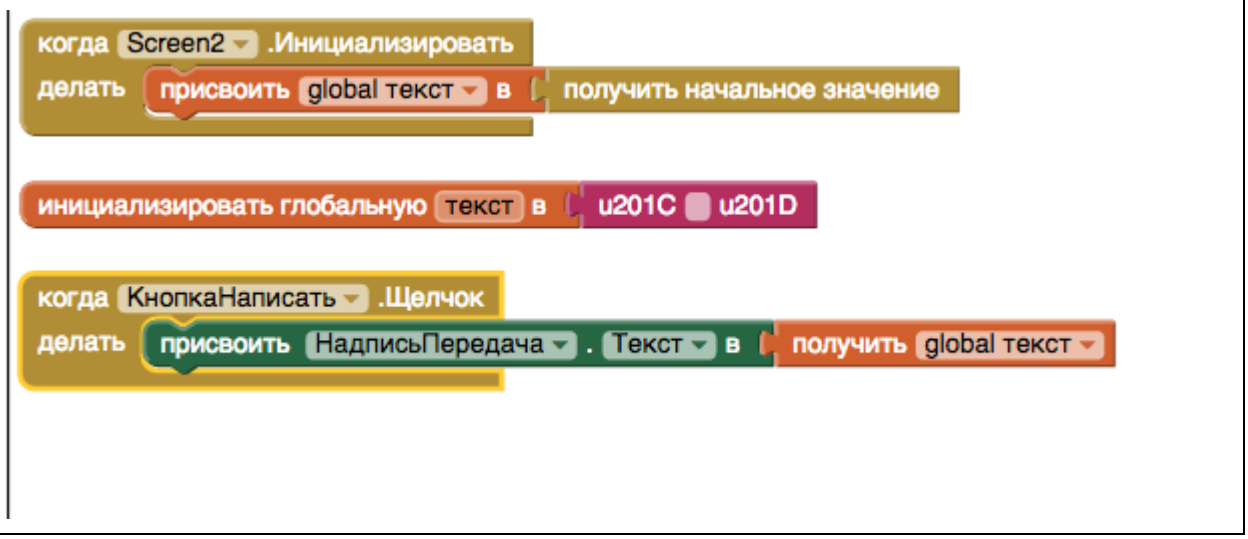

когда КнопкаГоворить .Щелчок
делать вызов РаспознавательРечи .ПолучитьТекст


когда КнопкаПередать .Щелчок
делать открыть ещё один экран с начальным значением названиеЭкрана u201C Screen2 u201D
начальноеЗначение РаспознавательРечи . Результат

Screen 2

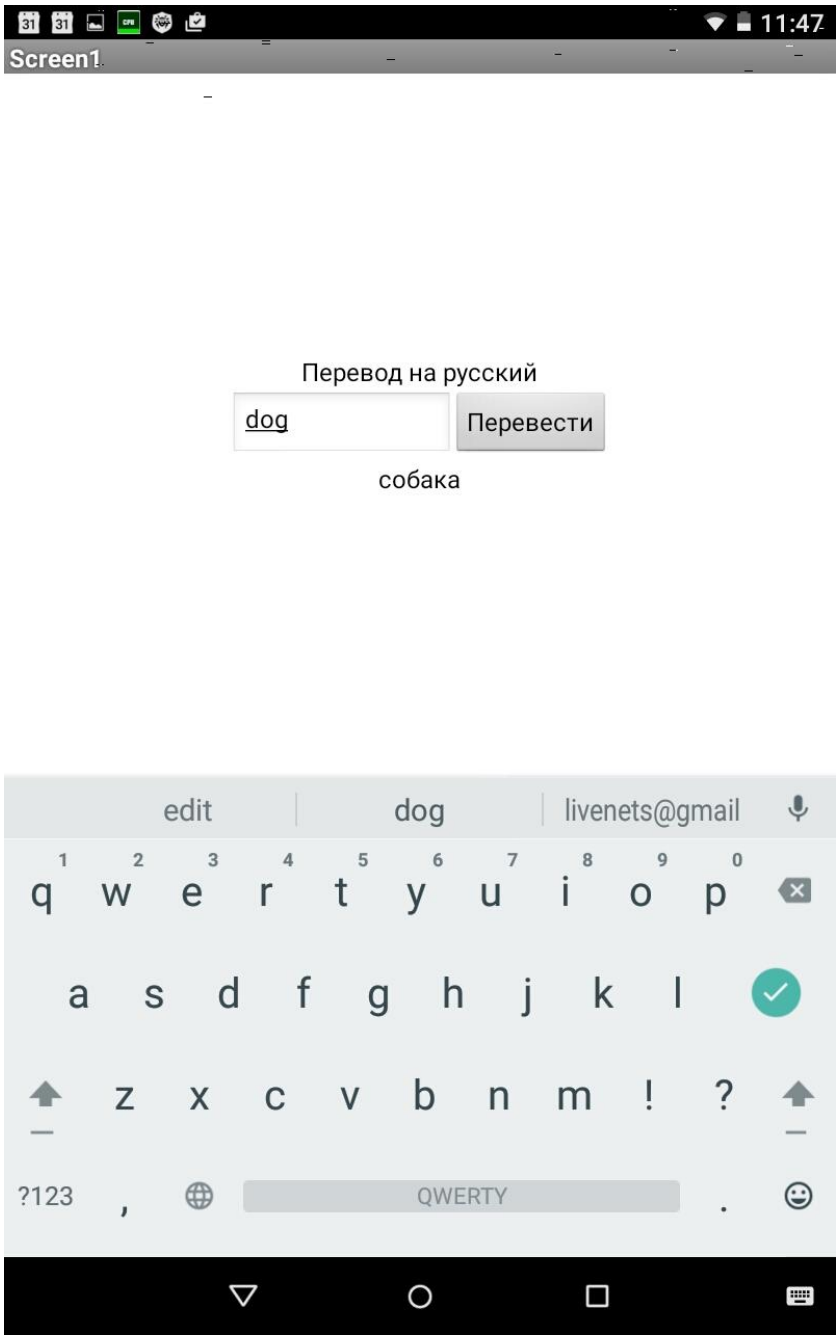

Screen2
Нажми, чтобы узнать что ты услышал

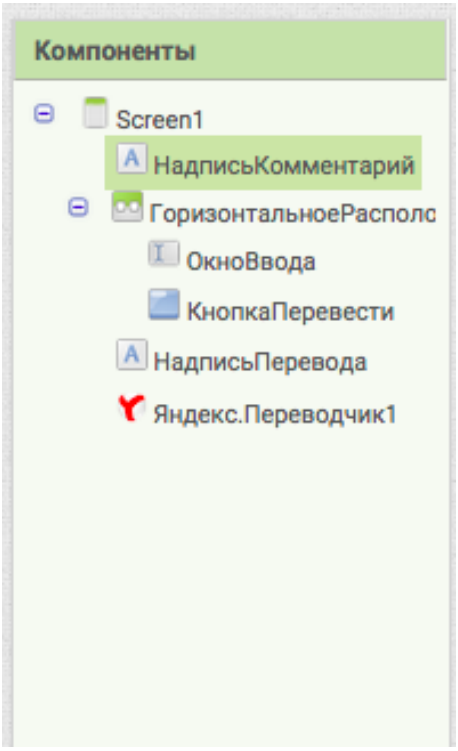

Компоненты
Screen2
ИзображениеЧеловек
КнопкаНаписать
НадписьПередача

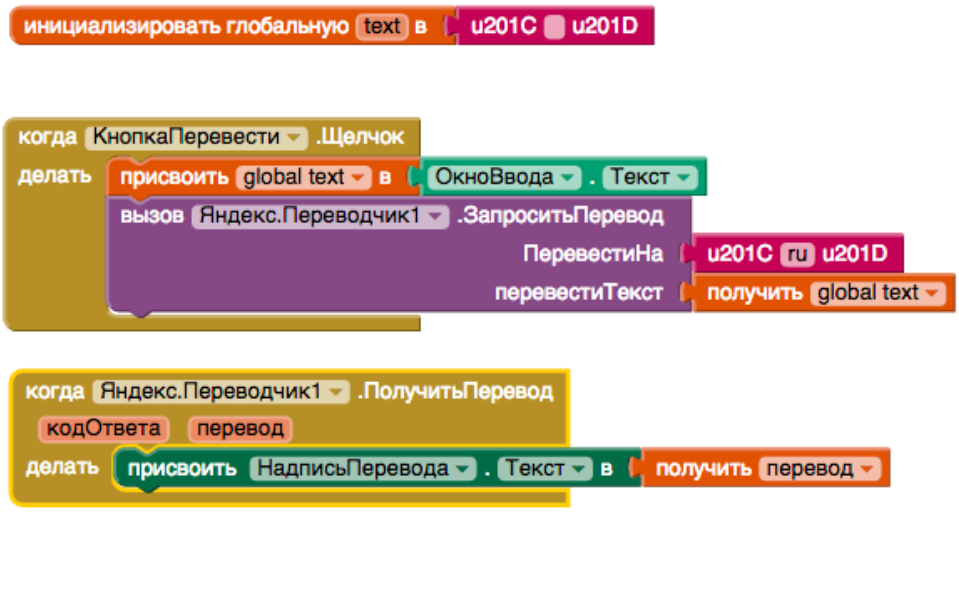

когда Screen2 .Инициализировать
делать присвоить global текст в получить начальное значение
инициализировать глобальную текст в u201C u201D
когда КнопкаНаписать .Щелчок
делать присвоить НадписьПередача . Текст в получить global текст

Пример 2.7.3 Приложение “Переводчик”

Описание: Приложение, которое переводит текст на другой язык

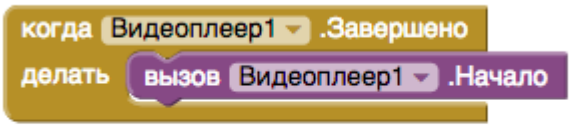


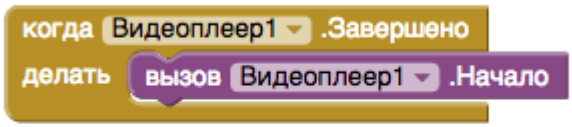



инициализировать глобальную text в u201C u201D
когда КнопкаПеревести .Щелчок
делать присвоить global text в ОкноВвода . Текст
вызов Яндекс.Переводчик1 .ЗапроситьПеревод
ПеревестиНа u201C ru u201D
переводитиТекст получить global text
когда Яндекс.Переводчик1 .ПолучитьПеревод
кодОтвета перевод
делать присвоить НадписьПеревода . Текст в получить перевод

Пример 2.7.4 Приложение “Видеоплеер”

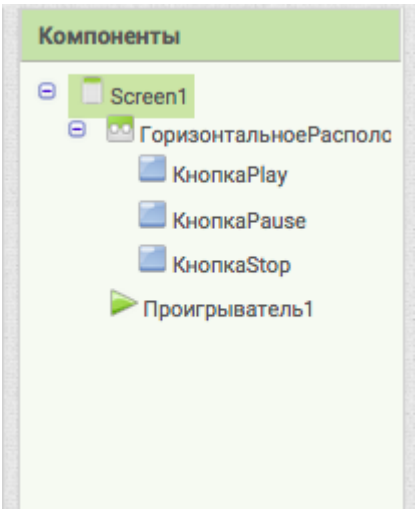
Описание. Приложение, которое проигрывает встроенный видеофайл.

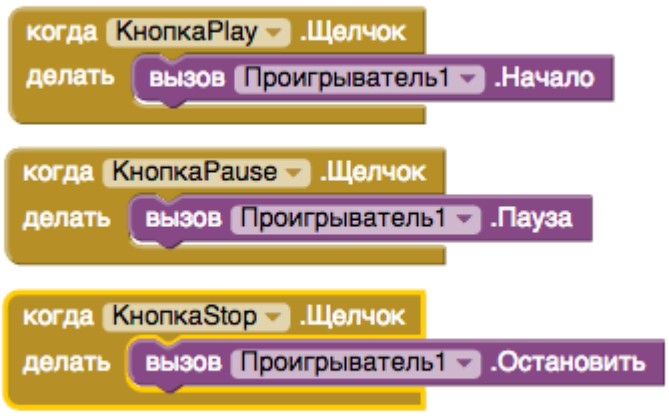




Пример 2.7.5 Приложение “Mp3 плеер”

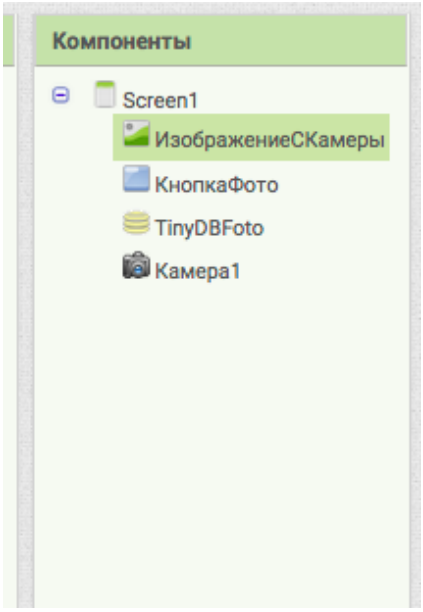
Описание. Приложение проигрывает звуковой файл, и реагирует на кнопки плеера.

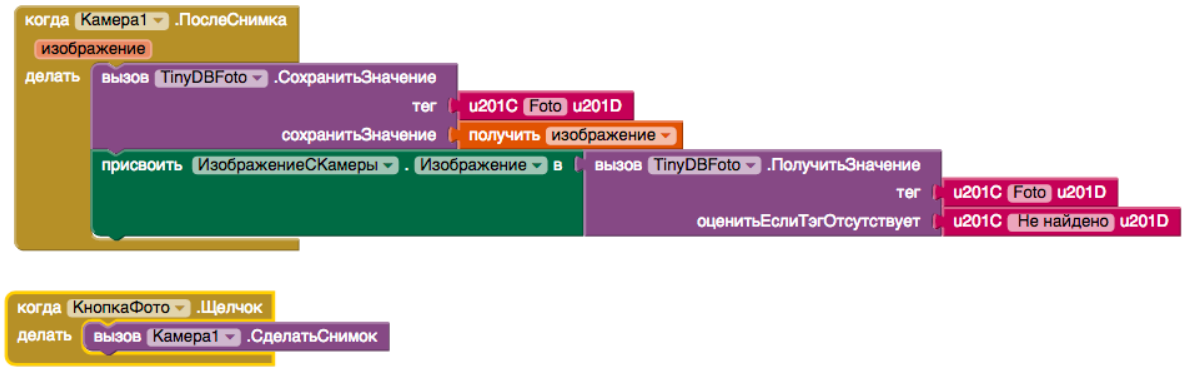




Пример 2.7.5 Приложение “Фотокамера”

Описание. Приложение позволяет сделать фото с встроенной камеры устройства и вывести его на экран приложения.





2.8 Общение

Общение - группа компонент, отвечающая за подключение к социальным сетям, открытия доступа к ресурсам, доступ к контактам телефона и пр.

- **СборщикКонтактов** представляет собой кнопку, при нажатии которой появляется возможность выбора контакта.

- **Сборщик Email** позволяет выбрать e-mail из списка контактов Android (до этого e-mail контакта должен быть там сохранен). Представляет собой текстовое поле, в который можно ввести или выбрать e-mail.

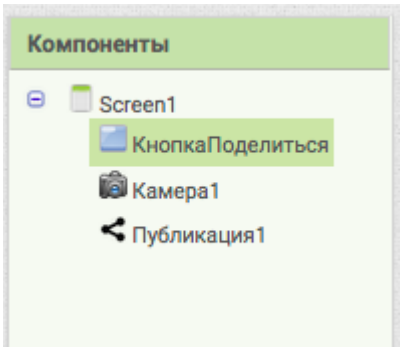
- **Позвонить** — невидимый компонент, который применяется для набора номера и совершения звонка.

- **Поделиться** - компонент, который позволяет обмениваться файлами и сообщениями между вашим приложением и другими приложениями, установленными на устройство. Этот элемент показывает список всех установленных приложений, которые могут обрабатывать информацию и позволяют пользователю выбрать то, с которым он хочет произвести обмен информацией, например, почтовое приложение, приложение социальной сети, обмена SMS и т.д.

- **Номеро-набиратель** позволяет выбрать номер телефона.

Пример 2.8.1 Приложение “Sharing”

Описание, Приложение, которое делает фото с камеры вашего мобильного устройства и публикует его в сети.



2.9 Сенсоры

Основные тезисы

Сенсоры - это микроустройства внутри смартфона или планшета, которые делают его умным, и связывают с внешним миром.

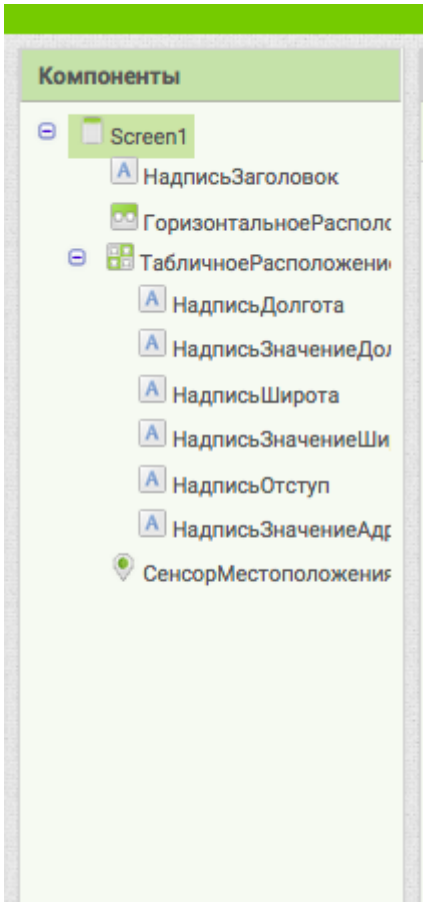
В мобильных приложениях могут быть использованы следующие сенсоры:

- Акселерометр (G-сенсор)— позволяет отследить ускорение, которое придается устройству, определяет ускорение устройства и используется в приложениях, где предполагается управлять действием приложения, изменяя положение устройства в пространстве, например, тряся его.
- Датчик приближения - позволяет определить приближение объекта без физического контакта с ним.
- Датчик освещенности позволяет определить степень наружного освещения и соответственно настроить яркость экрана.
- Гироскоп определяет положение в пространстве и позволяет отслеживать поворотом устройства и скорость поворота
- Магнитный компас (магнитометр) отслеживает ориентацию устройства в пространстве относительно магнитных полюсов Земли
- Сенсор ориентации обеспечивает получение данных от гироскопа и компаса мобильного устройства.
- Часы - обеспечивают отсчет времени, используя часы мобильного устройства Единица времени - миллисекунда. Позволяют отслеживать текущее время, считать временные промежутки и т.д.
- Сенсор местоположения получает данные от датчика GPS, определяя широту и долготу, а так же высоту над уровнем моря.
- При использовании сенсора местоположения GPS-датчик на устройстве должен быть включен.

Создание приложений, использующих сенсоры, требует предварительной проверки наличия сенсоров, имеющихся на устройстве.
К примеру приложение Sensor Box for Android , которое можно скачать из магазина Google Play обнаруживает все доступные датчики на устройства, и демонстрирует на графиках как они работают. Программы, написанные с использованием сенсоров, нужно тестировать не в эмуляторе, а на реальном устройстве, оснащенном данными датчиками.

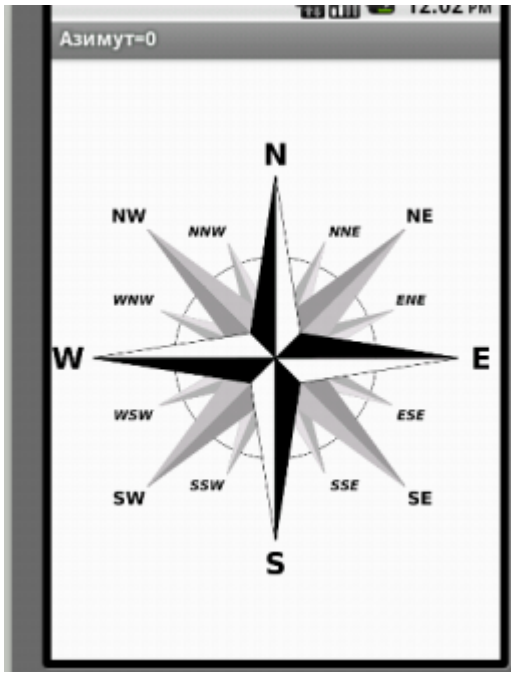
Пример 2.9.1 Приложение “Где я?”

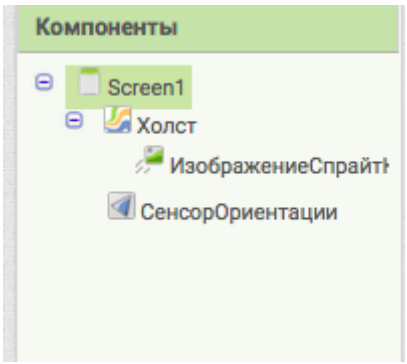
Описание: Приложение, которое выводит на экран широту, долготу и адрес местонахождения в настоящий момент.

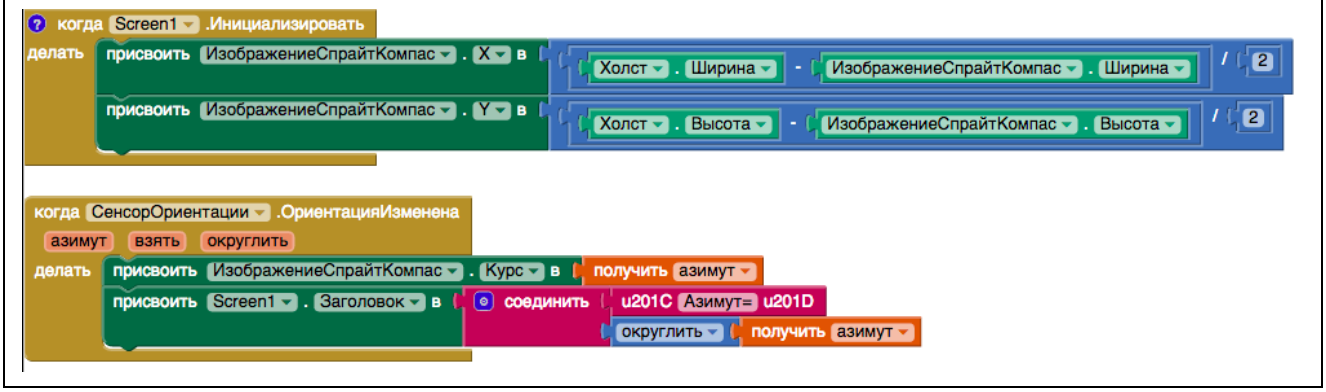


Пример 2.9.2 Приложение “Компас”

Описание: Приложение позволяющее определять стороны света.



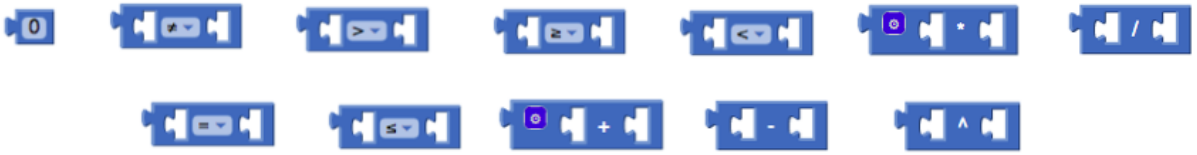




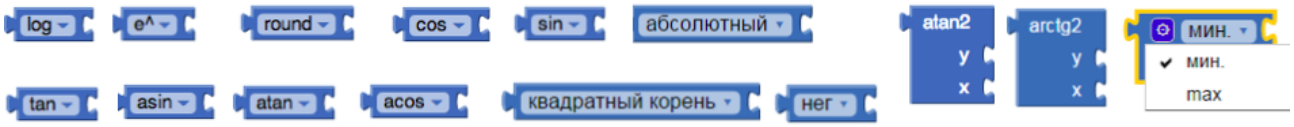
2.10 Математические функции

Группа блоков Математика содержит следующий набор операций и функций:

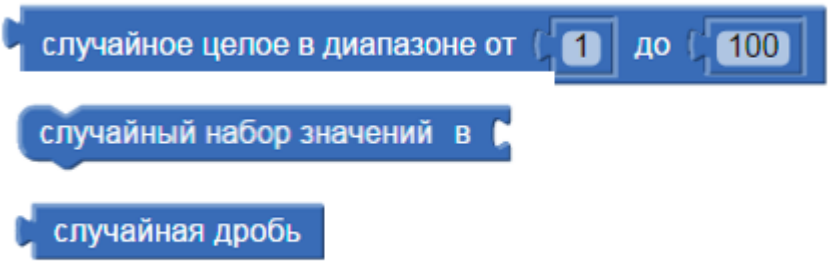
- математические операции (сложение, умножение, деление, вычитание, сравнения и др.)



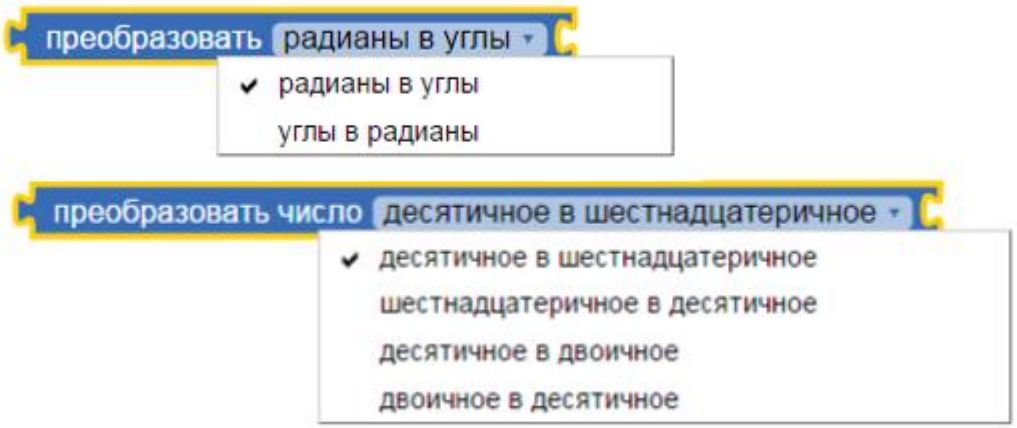
- математические функции



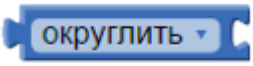
- функции случайных чисел

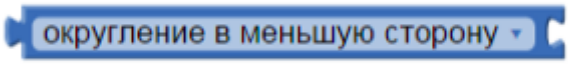


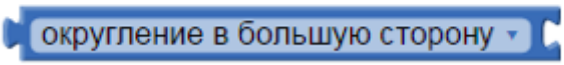
- функции преобразования одних величин в другие



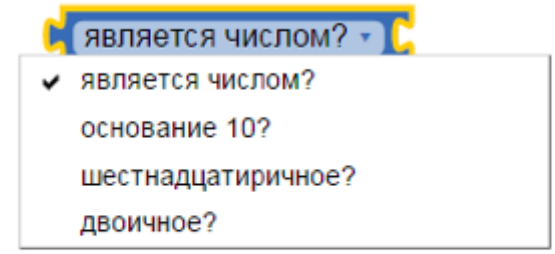
- функции округления величин

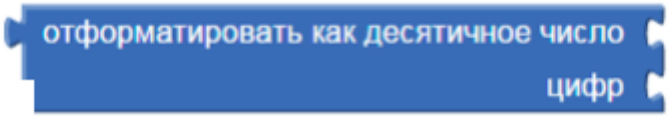




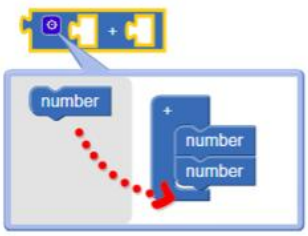


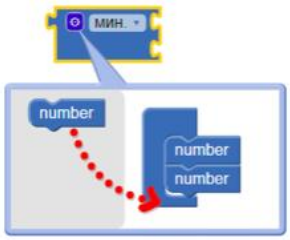
- работы с числами в различных системах счисления

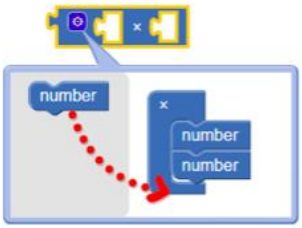




Получить доступ к различным функциям и построить сложные математические конструкции можно с помощью модификации блоков

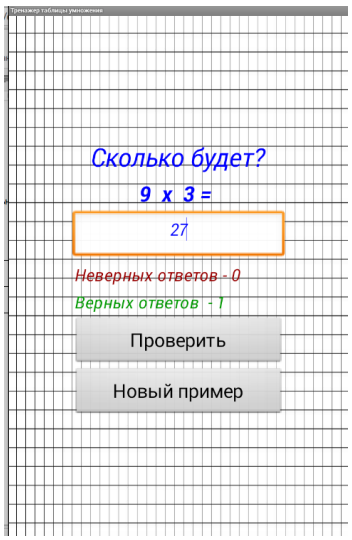


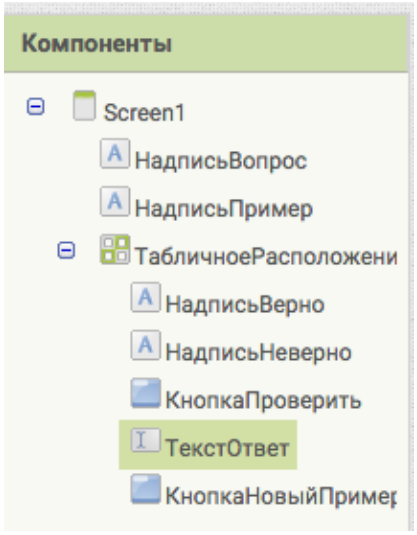


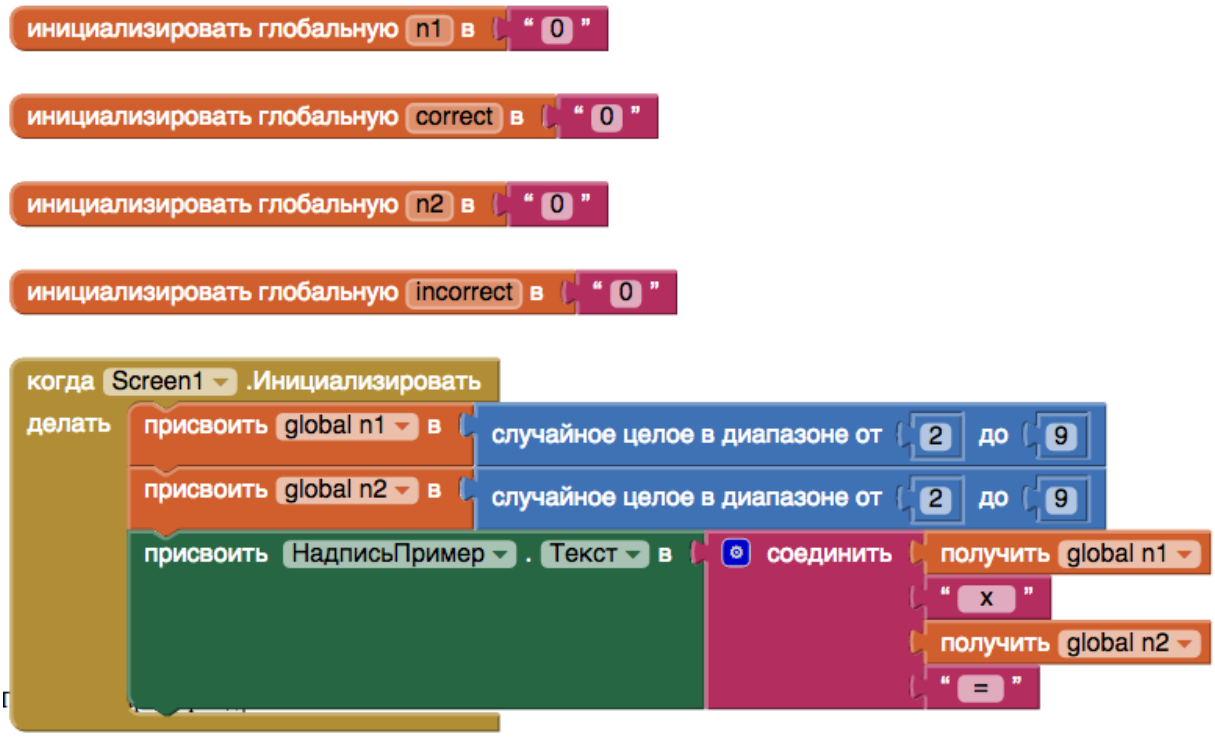


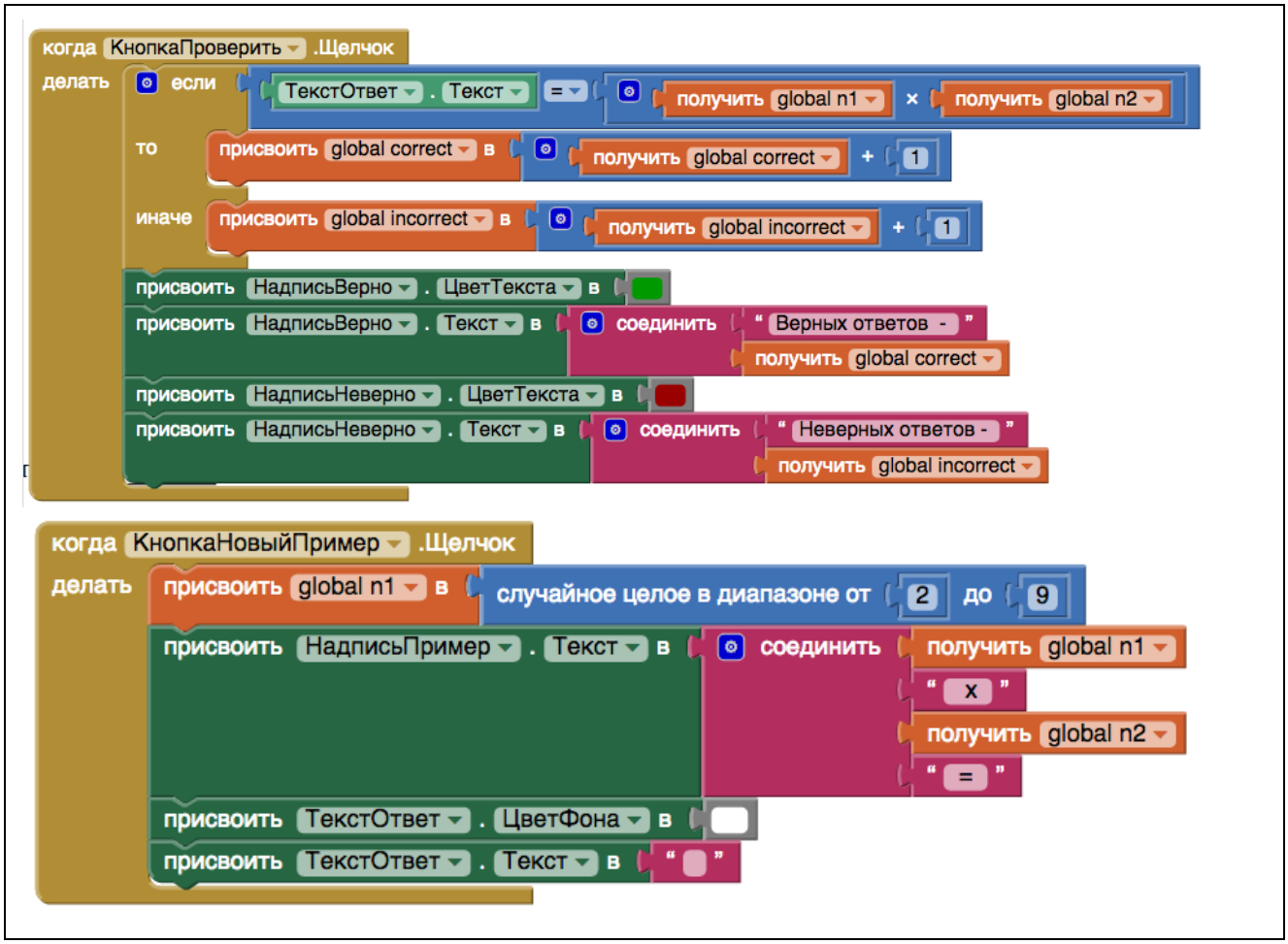
Пример 2.10.1. Приложение “Тренажер”

Описание. Приложение, которое проверяет правильность выполнения примеров на умножение однозначных чисел



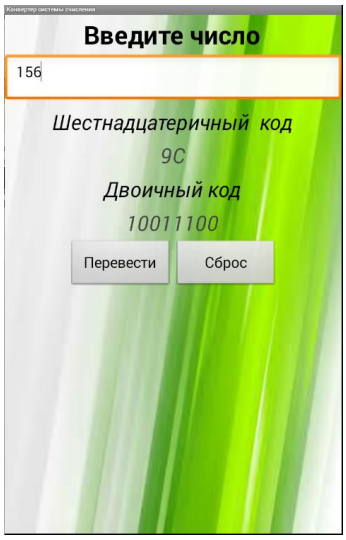


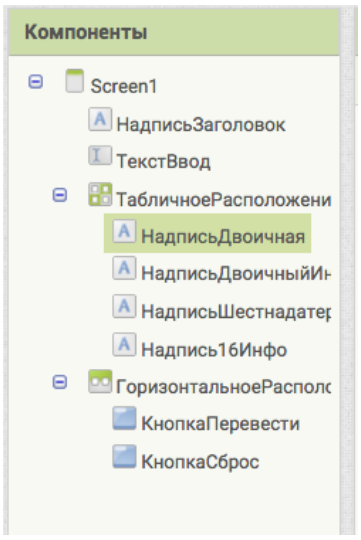


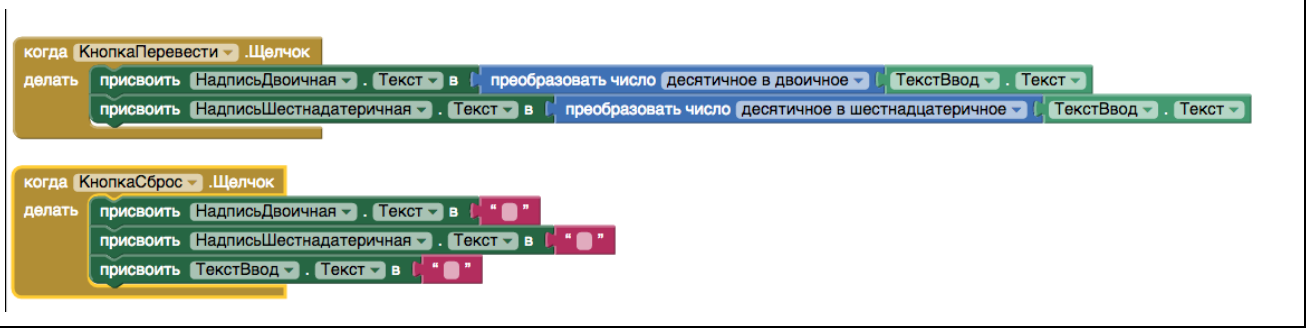

когда КнопкаПроверить .Щелчок
делать если ТекстОтвет . Текст = получить global n1 × получить global n2
то присвоить global correct в получить global correct + 1
иначе присвоить global incorrect в получить global incorrect + 1
присвоить НадписьВерно . ЦветТекста в
присвоить НадписьВерно . Текст в соединить " Верных ответов - " получить global correct
присвоить НадписьНеверно . ЦветТекста в
присвоить НадписьНеверно . Текст в соединить " Неверных ответов - " получить global incorrect
когда КнопкаНовыйПример .Щелчок
делать присвоить global n1 в случайное целое в диапазоне от 2 до 9
присвоить НадписьПример . Текст в соединить получить global n1 " x " получить global n2 " = "
присвоить ТекстОтвет . ЦветФона в
присвоить ТекстОтвет . Текст в " "

Пример 2.10.2. Приложение “Конвертер систем счисления”

Описание. Приложение, которое позволяет конвертировать введенные числа, в двоичную и шестнадцатеричную систему счисления.







Глава 3. Организация проектной деятельности

3.1 Совместная разработка приложений

Разработка мобильных приложений в среде MIT App Inventor может осуществляться не только одним человеком, но и целой командой

В этой ситуации возникает потребность проектирования и создания отдельных экранов разными людьми.

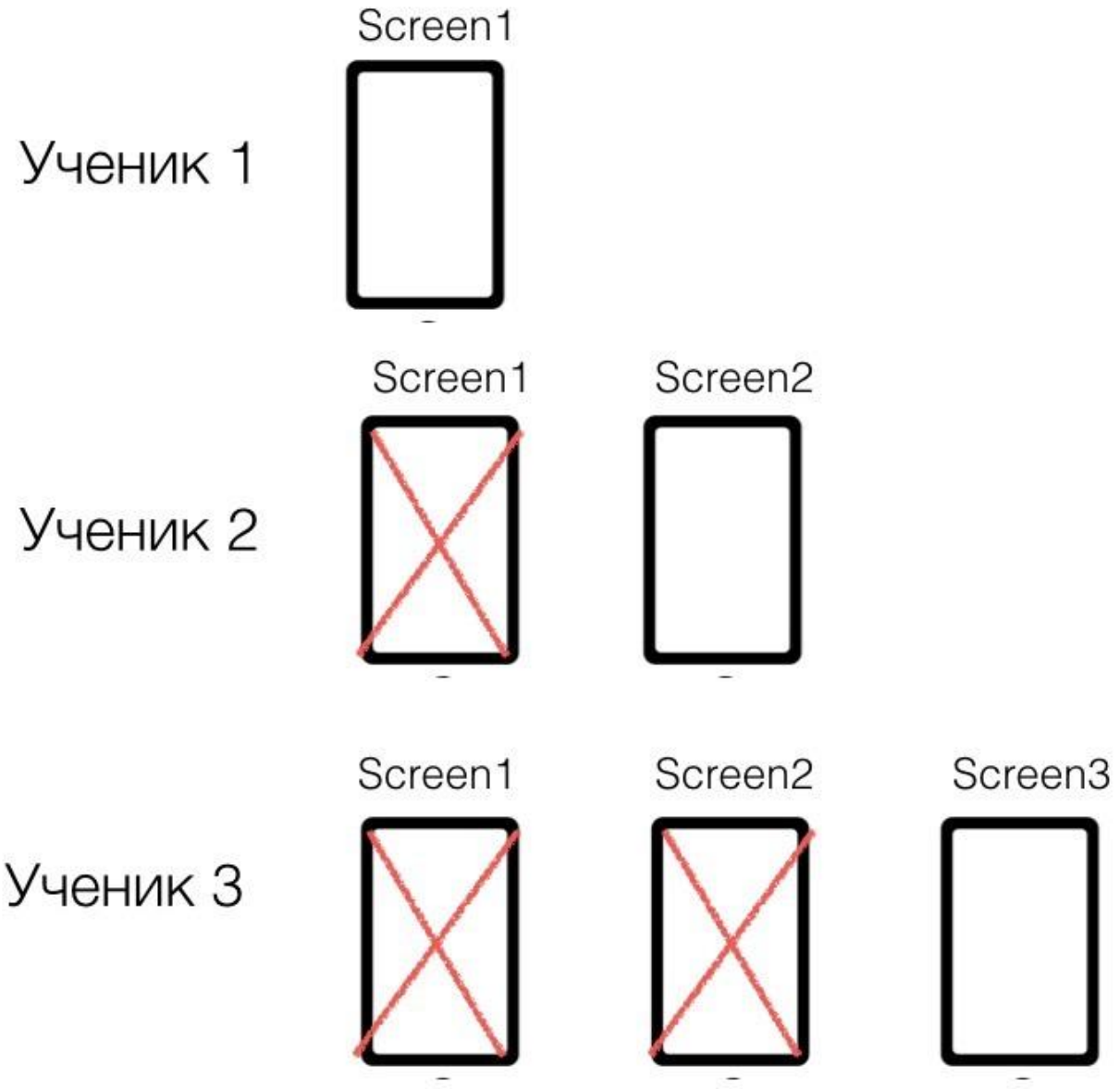
Структура программы в App Inventor привязана к “экранам”. У каждого экрана свой дизайн, свой набор кнопок, надписей, фонов и изображений, за каждым из них стоит своя программа, которую мы собираем в режиме просмотра блоков. Таких экранов в проекте App Inventor может быть сколько угодно. И возможность по организации совместной работы состоит в том, что мы можем собирать в один проект экраны, которые создают несколько программистов и дизайнеров.

Каждый участник совместной работы создает экран со своим номером
Screen1, Screen2 или ScreenN, оставляя экраны, создаваемые другими членами команды пустыми.

На финальном этапе работы выполняется слияние нескольких экранов в единое приложение с помощью инструмента AI2 Project Merger

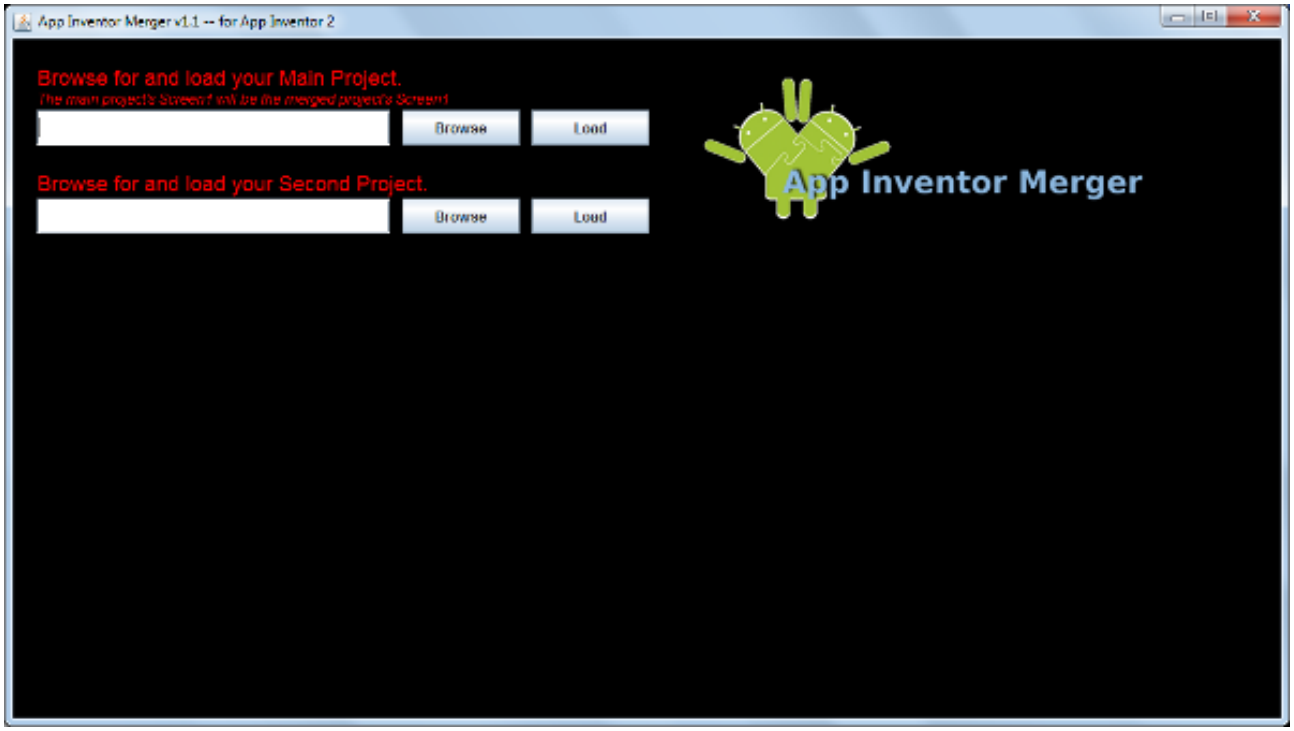
Алгоритм создания совместного приложения

1. Каждый участник создает отдельный проект.
2. Файлы проекта могут иметь одинаковые имена файлов или могут быть названы именами участников. Для того, чтобы минимизировать затраты на сбор приложения можно задавать имя проекта по схеме ИМЯ + Номер разрабатываемого экрана. Например m1, m2 и т. д.
3. Во всех разрабатываемых к проекту приложениях может быть только один экран Screen1, который нельзя переименовывать. Каждый экран создаваемый учеником может иметь только одну версию каждого экрана, которая должна быть в итоговом приложении.
4. Разработчик первого экрана заполняет только "Screen1." Все другие разработчики должны оставить "Screen1" пустым и разрабатывать дополнительные экраны.

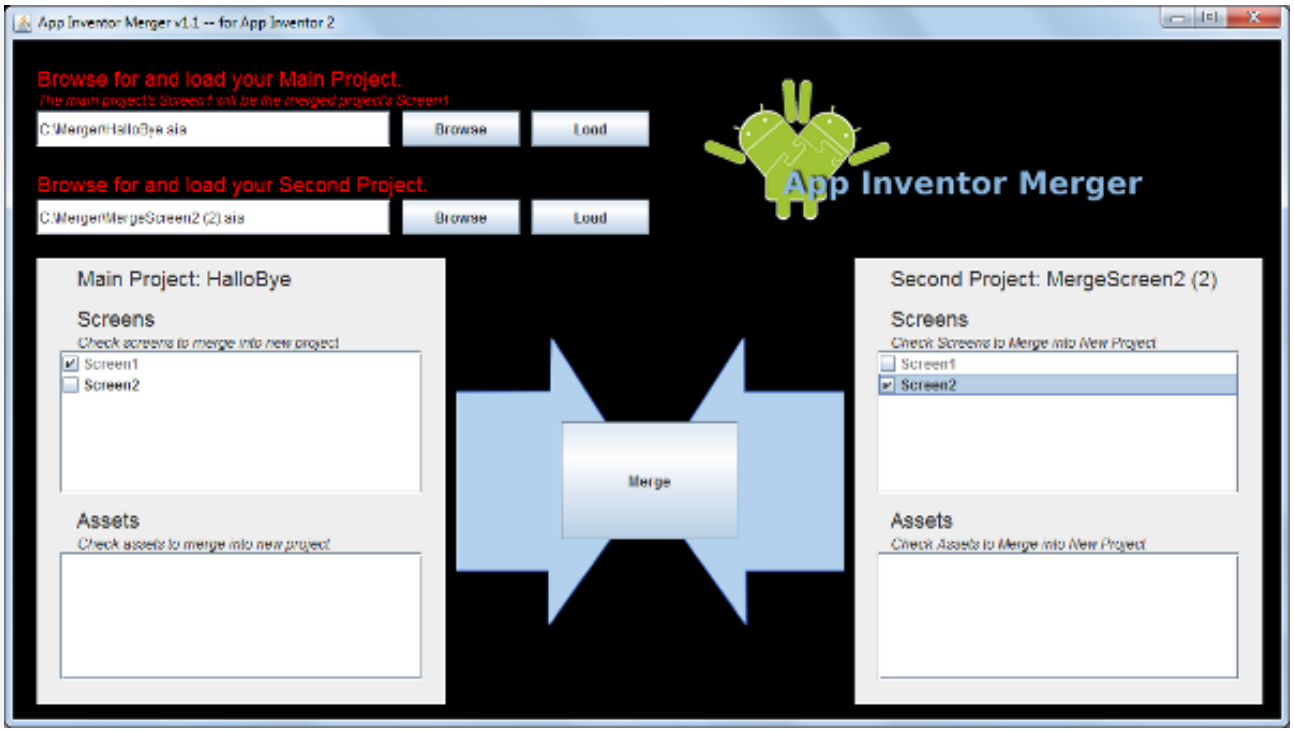


5. После того, как отдельные проекты завершены, их необходимо объединить вместе, используя App Inventor Merger Tool, загрузить его можно по ссылке http://appinventor.mit.edu/explore/resources/ai2-project-merger.html

6. Загрузить все созданные приложения как исходный код в формате .aia.



7. Загрузить исходные файлы экранов в соответствующие поля приложения App Inventor Merger.



8. Нажать на кнопку Merge. Программа генерирует aia файл, соединив в нем экраны.
9. Для добавления следующего проекта, шаги 6-8 нужно повторить.
10. Загрузить готовый проект на App Inventor

11. Настроить навигацию между экранами
12. Протестировать приложение на устройстве

3.2 Рекомендации к созданию итогового проекта - приложения

Этап 1. Анализ

Предложить несколько идей приложений. Проанализировать, кто может быть пользователями этих приложений, на кого оно ориентировано.

Где оно может быть использовано? Попробуйте сделать краткое описание приложения, действия, которые будут выполняться при его работе?

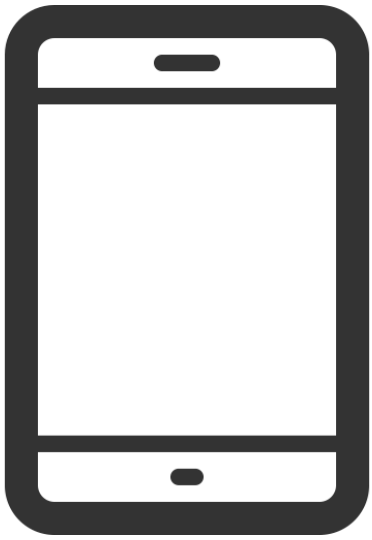
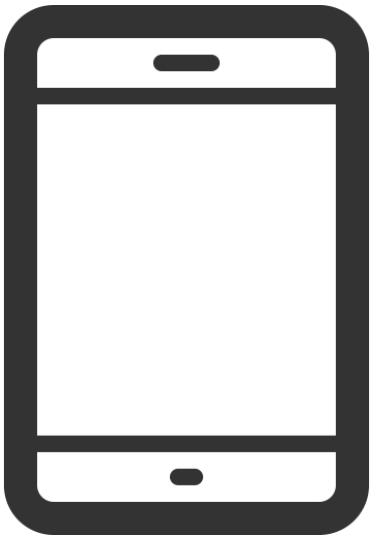
1._____

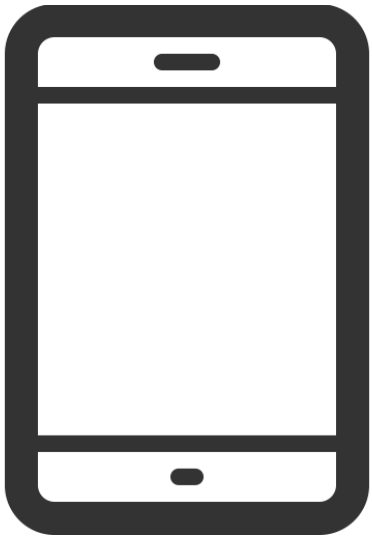
2._____

3._____

Этап 2. Разработка дизайна приложения

Разработка дизайна приложения включает разработку эскизов интерфейса пользователя для каждого экрана. На эскизе каждого экрана, желательно аннотировать, что делает каждый размещенный на нем компонент.

Screen 1	Screen2
	

Screen3	Пример
	

Этап 3. Проектирование действий для компонент приложения

Объясните, как приложение работает с точки зрения пользователя. Что произойдет, если пользователь касается экрана?

3.1 Перечислите , какие компоненты содержатся в программе?

- **видимые**

- **невидимые**

3.2 Вид: как эти компоненты отображаются при просмотре на мобильном устройстве?

_____

_____

_____

3.3. Мультимедиа. Какие мультимедийные файлы использованы в программе?

3.4. Сформулируйте какие свойства заданы для каждого из компонентов программы?

Событие представляет собой некоторое действие, которое активизирует стандартную реакцию компонента. В качестве события могут рассматриваться нажатие кнопки, выбор пункта меню.

Действия - операции или функции, которые деталь компонент может выполнять. Действия могут быть инициированы другими деталями посредством соответствующих связей.

Порядок выполнения действий определяется, прежде всего, событиями, возникающими в системе, и реакцией на них объектов..

При планировании приложений, рекомендуется создать таблицы, в которых прописаны используемые в приложении компоненты, свойства, события или действия.

Компонент	Действие	Как будет называться в панель компонент	Свойства	Что делает?
Кнопка	Перенести на экран	КнопкаДалее	Цвет фона: светло-серый Ширина - 80 пикселей Высота 50 пикселей	При нажатии переходит на экран Screen2

Этап 4. Программирование приложения в среде MIT App Inventor

Создать новый проект в среде MIT App Inventor. Создать дизайн приложения в режиме “Дизайн” и запрограммируйте действия для каждого компонента, в режиме “Блоки”.

Этап 5. Тестирование приложения

Проверить, как работает приложение, если возможно на устройствах с различными экранами. Составьте краткое описание вашего приложения.

Этап 6. Оценивание приложения

Провести оценку приложения на основании оценочного листа.

Название приложения	**Оценка в баллах (1,2,3)**	**комментарии**
Идея	**Простое**	
Дизайн приложения	**2**	
Программирование	**2**	
Внедрение	**2**	
Тестирование	**2**	
Краткое описание	**2**	

Этап 7. Оформление приложения

Разработать дизайн иконки, заставки или предусмотрите проигрывание звука, при запуске приложения, или запишите для него саундтрек.
Соблюдение этапов проектирования приложения, позволит сформировать у обучающихся навыки планирования и разработки приложений, которые помогут им в создании проектов.
Рассмотрим данный алгоритм на примере приложения “Аудиокнопка”.

Задание: Создать приложение в котором при нажатии на изображение проигрывается аудиофайл.

Этап 1. Анализ
Предложить несколько идей приложений. Проанализировать, кто может быть пользователями этих приложений, на кого оно ориентировано.
Где оно может быть использовано? Сделать краткое описание приложения, действия, которые будут выполняться при его работе?

1.Приложение может использоваться как напоминание выполнить какое то де
2.Приложение может работать как тест, при нажатии кнопки можно будет ответить на какой то вопрос.

3.Приложение может демонстрировать какие-либо действия, возникающие при программировании мобильных приложений.

Этап 2. Разработка дизайна приложения

Разработка дизайна приложения включает разработку эскизов интерфейса пользователя для каждого экрана. На эскизе каждого экрана, желательно аннотировать, что делает каждый размещенный на нем компонент.

Screen 1

Этап 3. Проектирование действий для компонент приложения

Объясните, как приложение работает с точки зрения пользователя. Что произойдет, если пользователь касается экрана?

При нажатии на Изображение (Кнопку) будет проигрываться звуковой файл.

3.1 Перечислите , какие компоненты содержатся в программе?

- **видимые**
  - *Кнопка*
  - *Надпись*
  - *Надпись*

- **невидимые**
  - *Звук*

3.2 Вид: как эти компоненты отображаются при просмотре на мобильном устройстве?

Надписи и кнопка изображения отображаются как только приложение запускается.

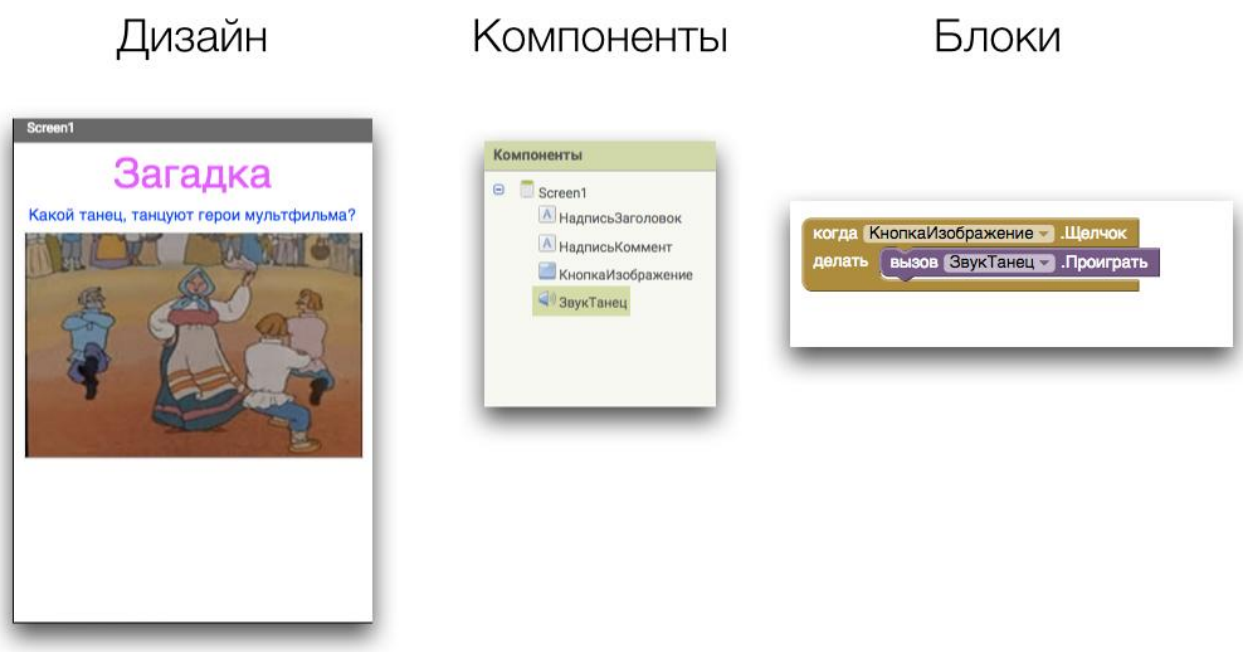
3.3. Мультимедиа. Какие мультимедийные файлы использованы в программе?
Звуковой файл dance. mp3

3.4. Сформулируйте, какие свойства заданы для компонентов программы?

Компонент	Тип компонента (Видимый/ Невидимый)	Как будет называться компонент?	Свойства	Событие	Действие
Экран Screen1	Видимый	По умолчанию Screen 1	Выравнивание по центру		
Надпись	Видимый	НадписьЗаголовок	Размер шрифта -40 px, цвет - пурпурный		
Надпись	Видимый	НадписьПодзаголовок	Размер шрифта - 16 px, цвет синий		
Кнопка	Видимый	КнопкаИзображение	Цвет фона: светло-серый Ширина - 300 пикселей Высота -200 пикселей Изображение Dance.jpeg	Нажатие кнопки	Вызов аудиофайла
Звук	Невидимый	ЗвукТанец	Источник Dance2.mp3.		Проигрывается при вызове

Этап 4. Программирование приложения в среде MIT App Inventor

Создайте новый проект в среде MIT App Inventor. Создайте дизайн приложения в режиме “Дизайн” и запрограммируйте действия для каждого компонента, в режиме “Блоки”.



Этап 5. Тестирование приложения

Проверьте как работает приложение, если возможно на устройствах с различными экранами. Составьте краткое описание вашего приложения.

Этап 6. Оценивание приложения

Проведите оценку приложения на основании оценочного листа.

Название приложения	**Оценка в баллах (1,2,3)**	**комментарии**
Идея	**Простая**	
Дизайн приложения	**2**	
Программирование	**2**	
Внедрение	**2**	
Тестирование	**2**	
Краткое описание	**2**	

Этап 7. Оформление приложения

Дизайн иконки на основе файла изображения. Санудтрека для запуска приложения нет.

Заключение

Данный практикум знакомит с основами программирования мобильных устройств в среде визуального программирования MIT App Inventor. Практические задания позволяют получить навыки создания мобильных приложений и оценить полезность и значимость развития навыков программирования для жизни. Выполнив задания практикума Вы сможете освоить основные принципы создания мобильных приложений. Научиться программировать можно только программируя. Успехов в создании приложений! Новых идей и новых программ!!

Литература

1. Kerfs J. Beginning Android Tablet Games Programming. – Apress, 2011. -198 c.
2. Frederick G., Lal R. Beginning Smartphone Web Development: Building Javascript, CSS, HTML and Ajax-Based Applications for iPhone, Android, Palm Pre, Blackberry, Windows Mobile and Nokia S60. – Apress, 2010. – 350 c.
3. Моррисон М. Создание игр для мобильных телефонов. – М.: ДМК Пресс, 2006. - 494 с.
4. Виноградов А. Программируем игры для мобильных телефонов. - М. –Триумф, 2007. – 272с.
5. MIT App Inventor. Ресурсы. http://appinventor.mit.edu/explore/resources.html
6. Я дилетант. Мобильные приложения своими руками http://idilettante.ru/category/mobilnye-prilozeniya/