



# Разработка и установка мобильного приложения



## Цели обучения:

- разработка мобильного приложения для мобильных устройств с использованием условных и циклических структур;
  - объяснить, как устанавливать разработанное приложение
- 

# Зачем нужны мобильные приложения?



# Когда нужно приложение



(клиенту)



ДОПОЛНИТЕЛЬНЫЕ  
ВОЗМОЖНОСТИ



новости | акции



доступ  
оффлайн



частота



скидка  
(в том числе  
оффлайн)



личный  
кабинет



(бренду)



данные



push-notification



лояльность



оффлайн  
интеграция



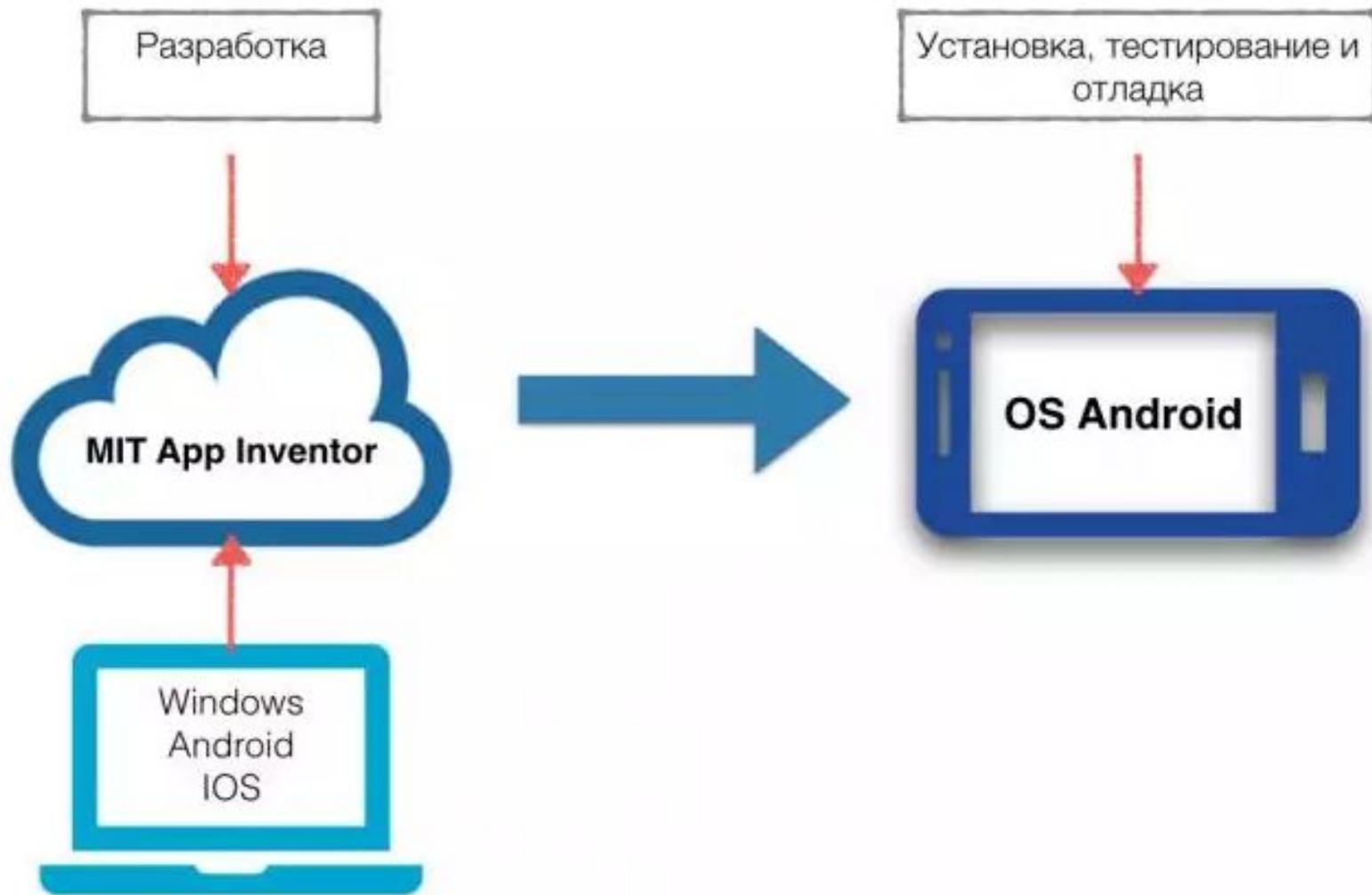
выше средний  
чек



## Когда НЕ нужно

- низкая частота покупки (например, b2b)
- нет денег и времени (поддержка и продвижение приложения)
- ассортимент не подходит для сторгов (например, товары 18+)

# Использование устройств





# 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.

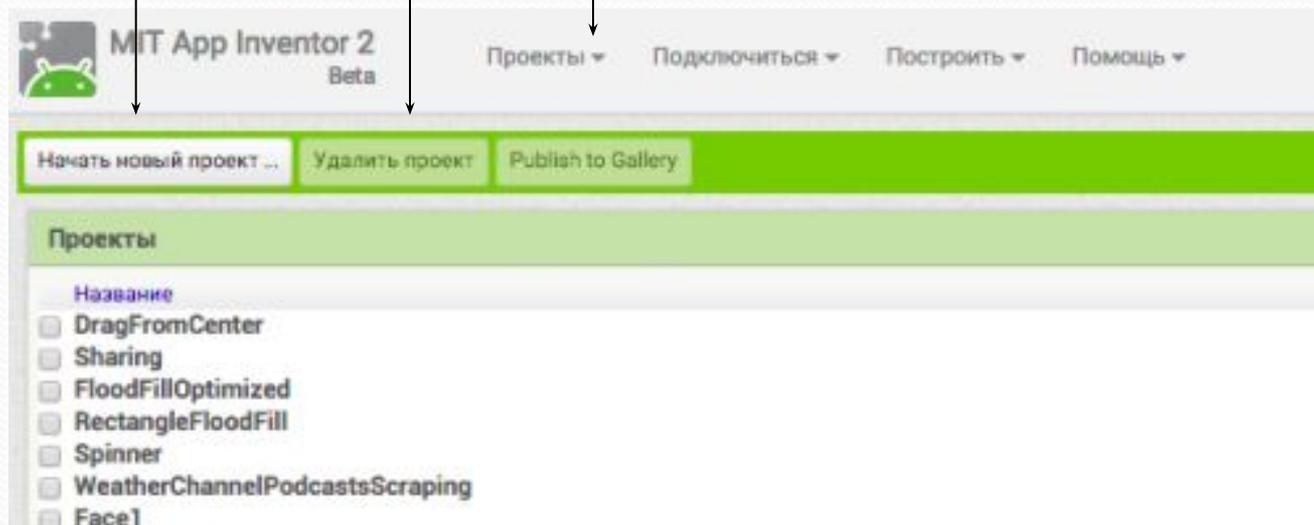


# Описание интерфейса пользователя

Вновь создаваемое  
приложение

Удаляет созданный  
проект

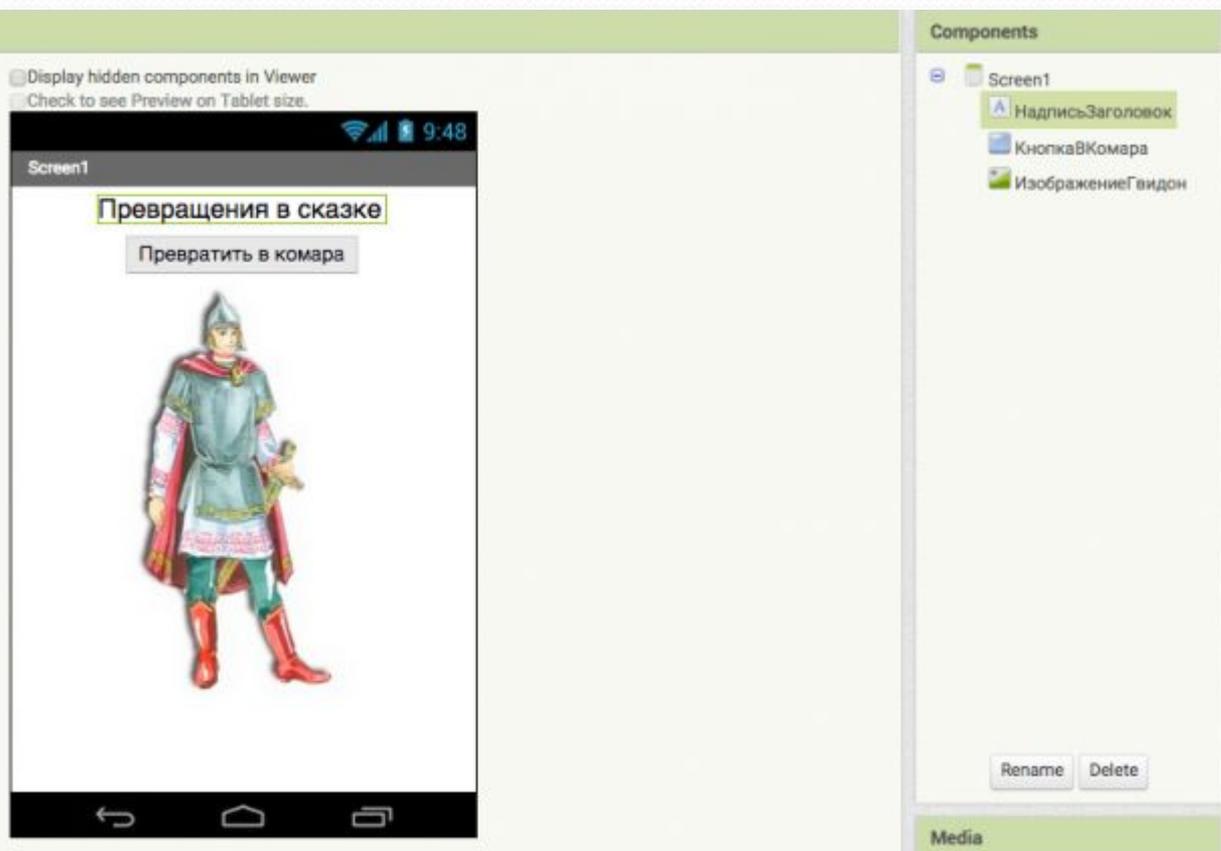
"Проекты/Мои проекты".  
Переход к списку проектов из окна  
разработки по ссылке



После входа в MIT App Inventor пользователь попадает на страницу, где отображается список его проектов.

Все созданные проекты хранятся в папке Мои проекты.

# Проектирование интерфейса ПОЛЬЗОВАТЕЛЯ



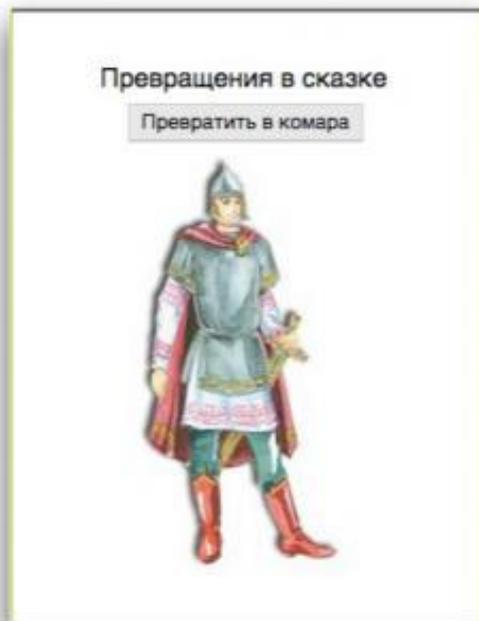
Режим "Дизайнер" - режим в котором создается интерфейс ("внешний вид") приложения. Данный режим используют для выбора и размещения различных компонент приложения: кнопок, текстовых полей, изображений и др., которые отображаются на экране вашего устройства, при запуске приложения.

# Экраны приложения

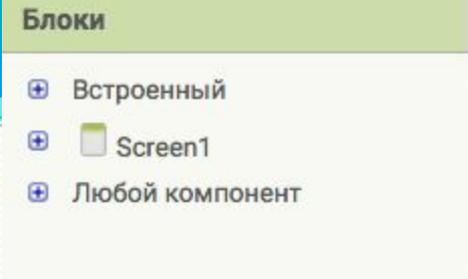
Проект приложение может состоять из множества экранов.

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.

## Найдем отличия



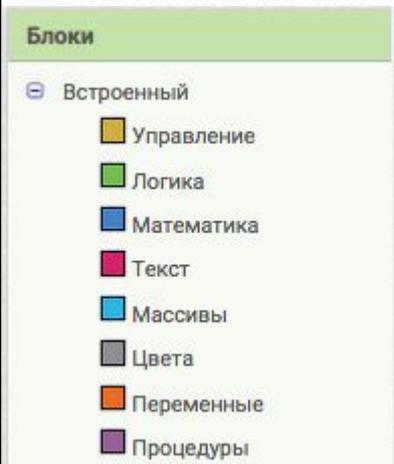
# Режим «Блоки»



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

## 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

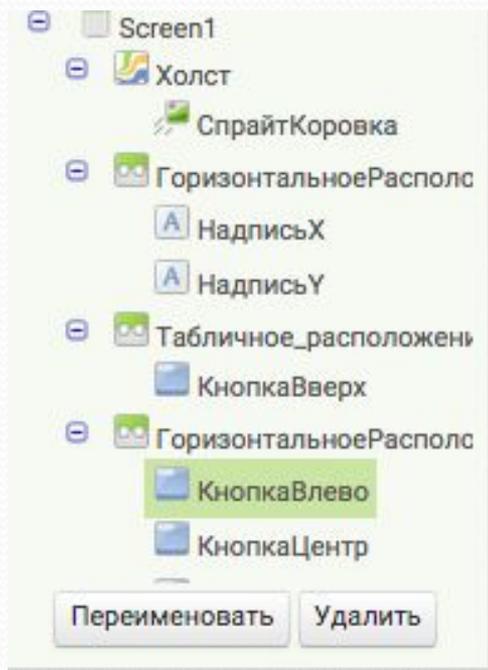
**Цвета** - определяет блоки по работе с цветами

**Переменные** - блоки позволяющие определять и устанавливать значение глобальных и локальных переменных

**Процедуры** - содержит блоки, позволяющие определять процедуры и функции

## 2. Блоки действий/событий для компонентов вашего приложения (Группа Screen 1)

Задаёт действия компонентам конкретного приложения. При выделении нужного компонента, отображаются доступные для него блоки



## 3. 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.



# Функции режима «Блоки»

При работе в режиме "Блоки" часто используются следующие его функции:  
**Свернуть/Развернуть блок.**



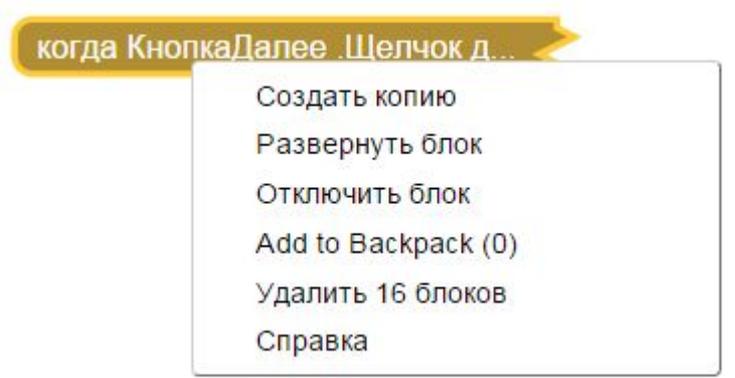
После выполнения функции "Свернуть блок", конструкция блоков принимает следующий вид:



## 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



# Функции режима «Блоки»

## Удалить блоки

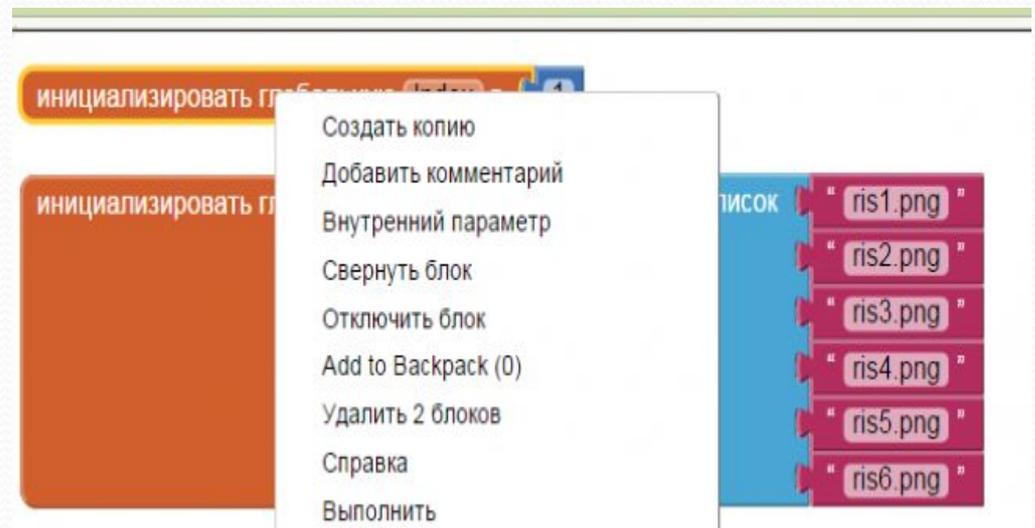
Блоки удаляются без перетаскивания в корзину

## Выполнить

Функция позволяет запустить к исполнению любую часть кода и используется при тестировании программ.

В этом случае необходимо иметь подключение к эмулятору.

## Копирование блоков



# 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.**

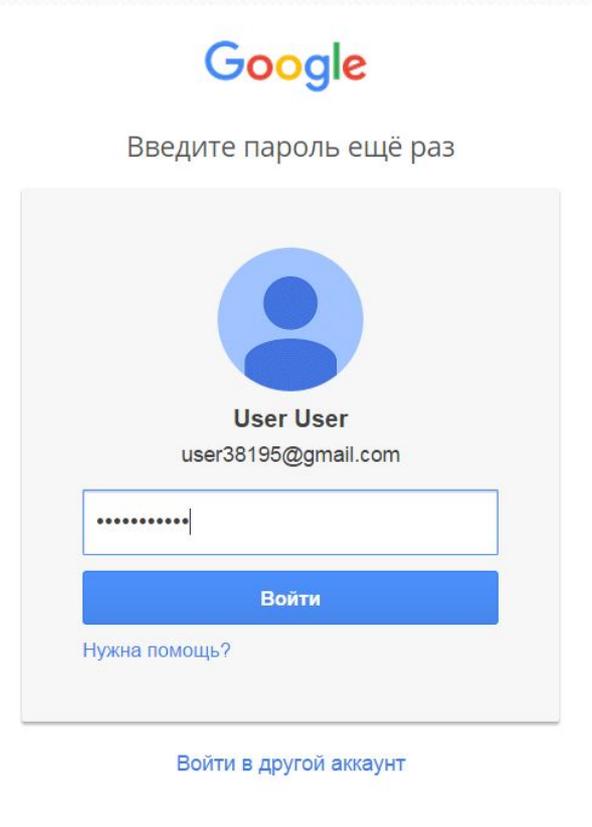
# Первое мобильное приложение



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”

Google Аккаунты

Приложение MIT AppInventor Version 2 запрашивает разрешение на доступ к вашему аккаунту Google.

Выберите аккаунт, который необходимо использовать.

user38195@gmail.com

Компания Google не связана с содержанием гостевой книги MIT AppInventor Version 2 или ее владельцами. Если вы выполните вход, компания Google отправит в гостевую книгу MIT AppInventor Version 2 ваш адрес электронной почты, но не пароль и личные данные.

Разрешить

Нет, спасибо

[Войти в д/](#)

To use App Inventor for Android, you must accept the following terms of service.

Помнить это подтверждение в течение 30 дней

©2011 Google - [Главна:](#)

## Terms of Service

### MIT App Inventor Privacy Policy and Terms of Use

*MIT Center for Mobile Learning*

Welcome to MIT's Center for Mobile Learning's App Inventor website (the "Site"). The Site runs on Google's App Engine service. You must read and agree to these Terms of Service and Privacy Policy (collectively, the "Terms") prior to using any portion of this Site. These Terms are an agreement between you and the Massachusetts Institute of Technology. If you do not understand or do not agree to be bound by these Terms, please immediately exit this Site.

MIT reserves the right to modify these Terms at any time and will publish notice of any such modifications online on this page for a reasonable period of time following such modifications, and by changing the effective date of these Terms. By continuing to access the Site after notice of such changes have been posted, you signify your agreement to be bound by them. Be sure to return to this page periodically to ensure familiarity with the most current version of these Terms.

#### Description of MIT App Inventor

From this Site you can access MIT App Inventor, which lets you develop applications for Android devices using a web browser and either a connected phone or emulator. You can also use the Site to store your work and keep track of your projects. App Inventor was originally developed by Google. The Site also includes documentation and educational content, and this is being licensed to you under the Creative Commons Attribution 4.0 International license ([CC BY 4.0](#)).

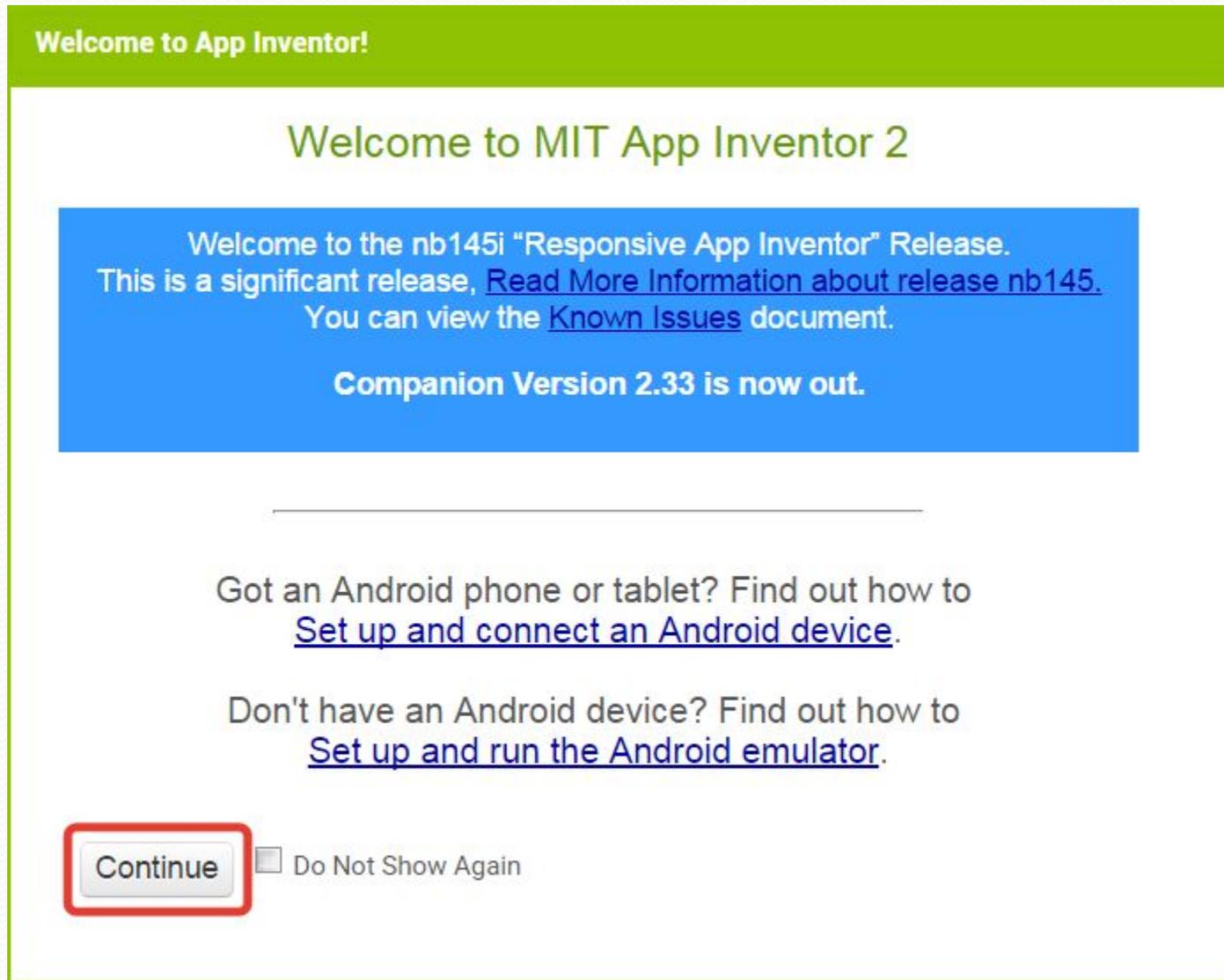
#### Account Required for Use of MIT App Inventor

In order to log in to MIT App Inventor, you need to use a Google account. Your use of that account is subject to Google's Terms of Service for

I accept the terms of service!

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->Название проекта”

# Практическая работа

## 1. «Виртуальный кот»

На экран выходит кот, который издает звук, когда его погладят.



The image shows two parts of a software development environment. On the left is a preview window titled 'Screen1' containing a cartoon illustration of a blue and white striped cat with its mouth open as if meowing. Below the cat is the text 'Мой кот'. On the right is a component tree titled 'Компоненты' (Components) showing a hierarchy: 'Screen1' (with a minus sign icon) contains three sub-components: 'КнопкаКот' (ButtonCat), 'НадписьКот' (LabelCat), and 'ЗвукМяу' (SoundMeow).

когда **КнопкаКот** .ПровестиВниз  
делать **вызов ЗвукМяу** .Проиграть

# Задание 2 «Хамелеон».

## Приложение «Хамелеон» Интерфейс

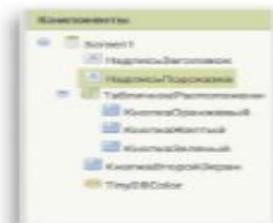
Screen 1.  
До выбора цвета



Screen 1.  
После выбора цвета



### Компоненты



```
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
```

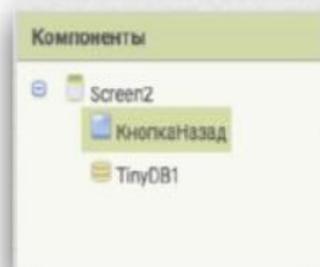
```
когда Screen1 .Инициализировать  
делать  
    вызов TinyDBColor .ОчиститьВсё  
    присвоить Screen1 . ЦветФона в 
```

# Интерфейс



# Screen 2.

# Компоненты



# Блоки

```
когда КнопкаНазад .Щелчок
  делать открыть другой экран названиеЭкрана u201C Screen1 u201D

когда Screen2 .Инициализировать
  делать присвоить Screen2 . ЦветФона в вызов TinyDB1 .ПолучитьЗначение
    тег u201C Color u201D
    оценитьЕслиТэгОтсутствует
```

# Формативное задание

- 1) Назовите типы расположения элементов интерфейса.
- 2) Укажите, какие из следующих компонентов отображаются на экране, а какие нет.



Кнопка



Звук



Проигрыватель



Текст

- 3) Назовите список компонентов необходимых для создания приложения, которое проигрывает звук при нажатии на кнопку.

## Приложение для туристов «Слайд-шоу Достопримечательности Казахстана»

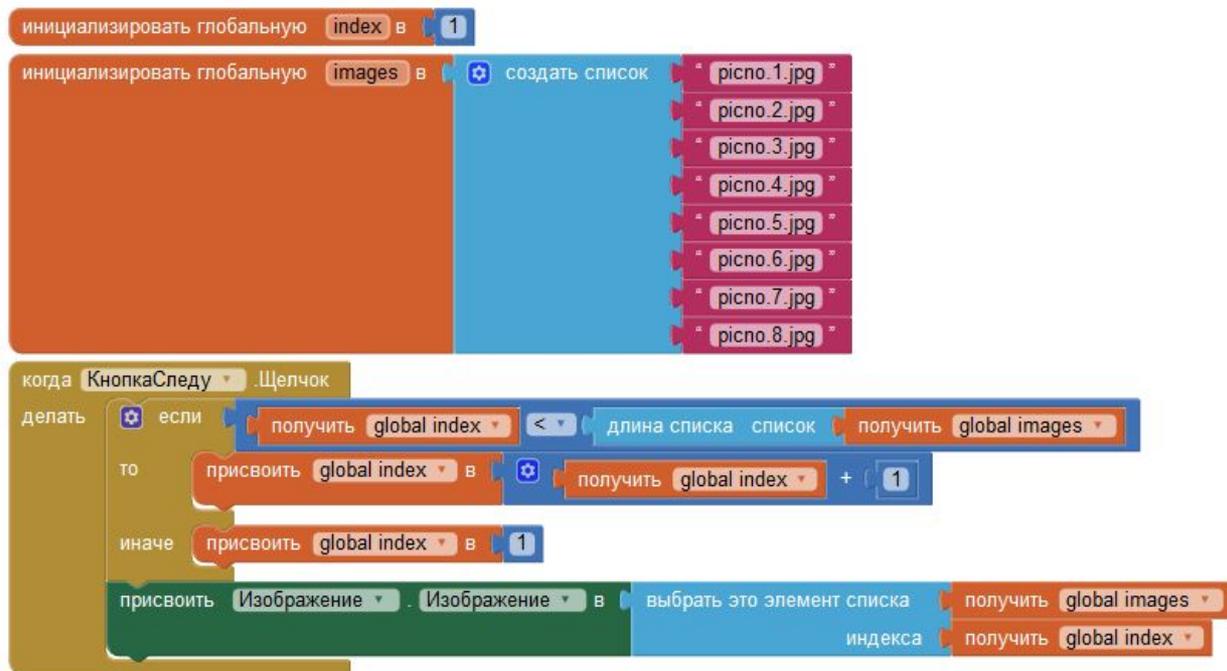
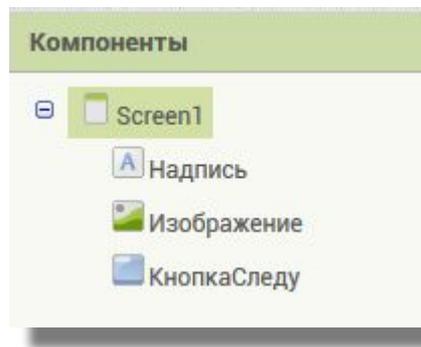
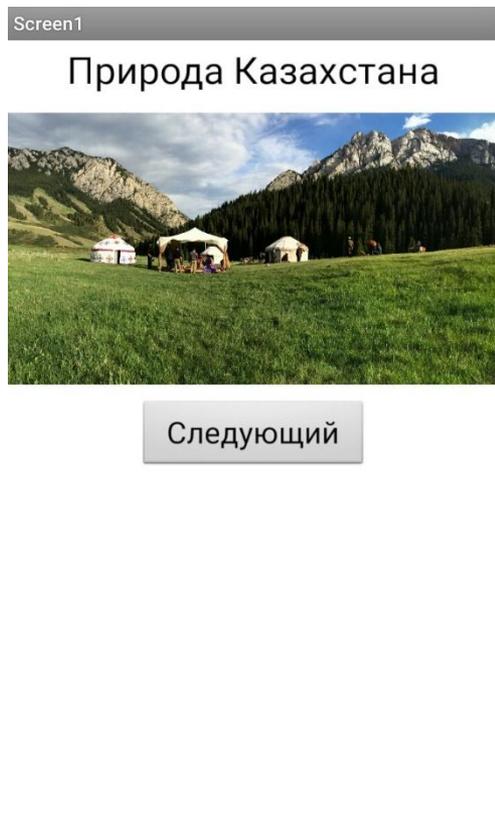
Создайте приложение для туристов нашей страны, которые могли бы ознакомиться в виде слайд-шоу с достопримечательностями Казахстана.



# Практическая работа

## 1. “Слайд-шоу”

Приложение, отображающее слайд-шоу из изображения.



# Учебный материал

Способы загрузки приложения на устройство:

- *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)

