



Introduction to AppInventor

Alark Joshi, Amit Jain, Sydney Crabtree

What is AppInventor?



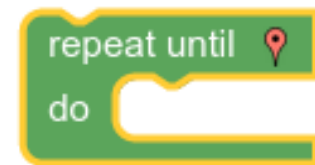
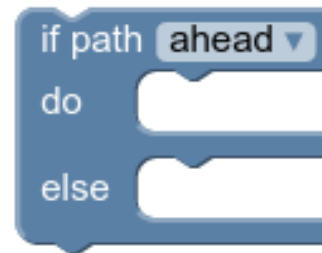
- AppInventor is a visual “blocks” programming language
- Allows you to create mobile apps for the Android platform

Why AppInventor?

- Promotes creativity and eliminates annoying details
- Engaging and exciting
- Ability to share your app with anyone in the world
- Ideal for teaching an introductory computer science course to middle-school and high-school students

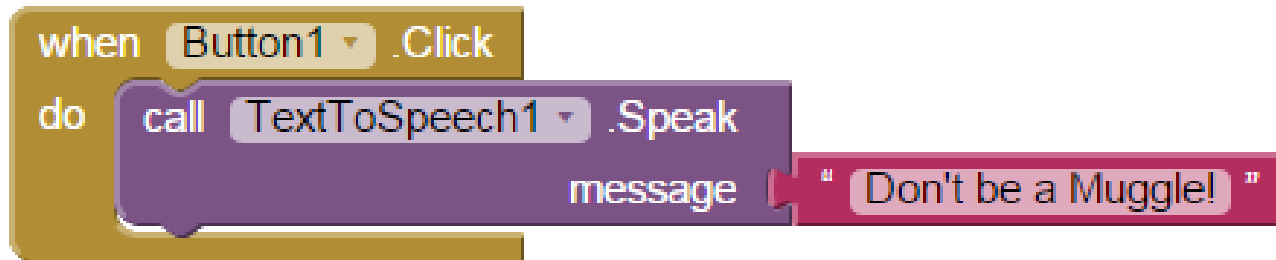
Blocks programming language

- AppInventor is a “blocks programming” language
- What is “blocks programming”?
 - Similar to Blockly
- What was interesting/engaging about Blockly?




“Blocks” programming

- Actions are clear



```
when Button1 .Click  
do call TextToSpeech1 .Speak  
message " Don't be a Muggle! "
```

A Scratch-style code block with a gold header containing the text "when Button1 .Click". Below the header is a purple block labeled "do" containing a call block "call TextToSpeech1 .Speak" and a message block "message" with a pink tail containing the text " Don't be a Muggle! ".



```
when AccelerometerSensor1 .Shaking  
do call TextToSpeech1 .Speak  
message " Stop shaking me! "
```

A Scratch-style code block with a gold header containing the text "when AccelerometerSensor1 .Shaking". Below the header is a purple block labeled "do" containing a call block "call TextToSpeech1 .Speak" and a message block "message" with a pink tail containing the text " Stop shaking me! ".

“Blocks” programming

- All possible actions are clearly listed

The image shows a visual programming environment with two main panels: 'Blocks' and 'Viewer'.

Blocks Panel:

- Built-in:**
 - Control (Yellow)
 - Logic (Green)
 - Math (Blue)
 - Text (Red)
 - Lists (Light Blue)
 - Colors (Grey)
 - Variables (Orange)
 - Procedures (Purple)
- Screen1:**
 - Button1
 - TextBox1
 - TextToSpeech1** (highlighted)
 - AccelerometerSensor1
- Any component**

Buttons: Rename, Delete

Media Panel:

- Upload File ...

Viewer Panel:

The viewer displays a sequence of blocks for a `TextToSpeech1` component:

- `when TextToSpeech1 .AfterSpeaking`
 - `result`
 - `do` (message block)
- `when TextToSpeech1 .BeforeSpeaking`
 - `do` (call `TextToSpeech1 .Speak` message block)
- `call TextToSpeech1 .Speak` (message block)
- `TextToSpeech1 . AvailableCountries`
- `TextToSpeech1 . AvailableLanguages`
- `TextToSpeech1 . Country`
- `set TextToSpeech1 . Country to`
- `TextToSpeech1 . Language`
- `set TextToSpeech1 . Language to`
- `TextToSpeech1 . Pitch`
- `set TextToSpeech1 . Pitch to`
- `TextToSpeech1 . Result`
- `TextToSpeech1 . SpeechRate`
- `set TextToSpeech1 . SpeechRate to`
- `TextToSpeech1`

“Blocks” programming

- Errors are **mostly avoided** due to the nature of the blocks
- Built-in modules for fast development
 - Go from an idea to an app quickly
- Easy and fun to make your own apps!

What can you build with AppInventor?

Games

Android, Where's my car?



Text Processing Apps



Quizzes

Web Enabled Apps

Location Aware Apps

Broadcast Hubs

AppInventor Components

- AppInventor has the following three components:
 1. Component Designer (Web-based)
 2. Blocks Editor (Web-based)
 3. Emulator/Mobile Device

AppInventor 2: Component Designer

App Inventor 2 Beta

Projects ▾ Connect ▾ Build ▾ Help ▾ My Projects Gallery Guide Report an Issue English ▾ ajain@boisestate.edu ▾

TalkToMe Screen1 ▾ Add Screen ... Remove Screen **Designer** Blocks

Palette

User Interface

- Button
- TextBox
- ListView
- DatePicker
- TimePicker
- CheckBox
- Label
- ListPicker
- Slider
- PasswordTextBox
- Notifier
- Image
- WebView
- Spinner

Layout

Media

Drawing and Animation

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Viewer

Display hidden components in Viewer

Screen1

Talk To Me

Non-visible components

TextToSpeech1 AccelerometerSensor1

Components

- Screen1
 - Button1
 - TextBox1
 - TextToSpeech1
 - AccelerometerSensor1

Rename Delete

Media

Upload File ...

Properties

Screen1

AboutScreen

AlignHorizontal: Left

AlignVertical: Top

AppName: TalkToMe

BackgroundColor: White

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

ScreenOrientation: Unspecified

Scrollable:

Title: Screen1

VersionCode: 1

AppInventor 2: Blocks Editor

The screenshot displays the MIT App Inventor 2 Beta interface. At the top, the navigation bar includes the MIT App Inventor 2 logo, the text "MIT App Inventor 2 Beta", and menu items for "Projects", "Connect", "Build", and "Help". On the right side of the navigation bar, there are links for "My Projects", "Gallery", "Guide", "Report an Issue", "English", and the user email "ajain@boisestate.edu".

The main workspace is titled "TalkToMe" and features a green header bar with "Screen1" selected, "Add Screen ..." and "Remove Screen" buttons, and "Designer" and "Blocks" tabs. The interface is divided into three main sections:

- Blocks:** A sidebar on the left containing a "Built-in" category with sub-categories like Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Below this is a tree view for "Screen1" showing components: Button1, TextBox1, TextToSpeech1, and AccelerometerSensor1. At the bottom of the sidebar are "Rename" and "Delete" buttons, and a "Media" section with an "Upload File ..." button.
- Viewer:** A central area showing a list of event-driven blocks for Button1 and AccelerometerSensor1. The blocks include: "when Button1 .Click", "when Button1 .GotFocus", "when Button1 .LongClick", "when Button1 .LostFocus", "when Button1 .TouchDown", "when Button1 .TouchUp", "Button1 . BackgroundColor", "set Button1 . BackgroundColor to", "Button1 . Enabled", "set Button1 . Enabled to", "Show Warnings", and "Button1 . FontBold".
- Code Editor:** A large white area on the right where the selected blocks are assembled into code. It shows two event blocks: "when Button1 .Click" with a "do" block containing "call TextToSpeech1 .Speak" and "message 'Don't be a Muggle!'", and "when AccelerometerSensor1 .Shaking" with a "do" block containing "call TextToSpeech1 .Speak" and "message 'Stop shaking me!'". A trash can icon is located in the bottom right corner of the code editor.

Lets make an App

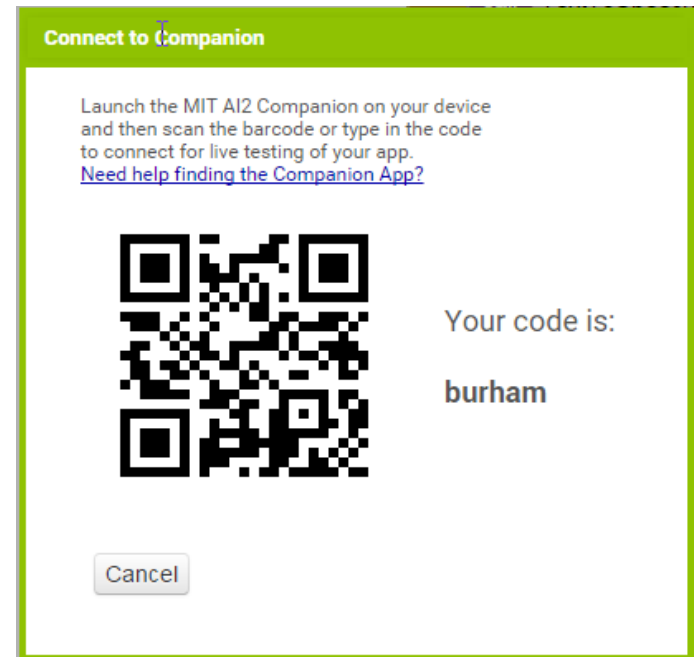
- Talk to Me 😊
 - Version 1
 - A *button*, when clicked says a pre-canned message
 - Version 2
 - Add a *textbox* that allows the user to type the message that should be spoken by the device
 - When the device is shaken, it admonishes the user

Getting Started

- You need a Gmail account
- You need Chrome or Firefox (IE is not supported at present)
- Go to <http://ai2.appinventor.mit.edu/> and sign in using your Gmail account
- Install the free MIT AI2 Companion app on your Android device

Connecting your Android device over WiFi

- Select Connect → AI Companion



- Start the MIT AI2 Companion app and either scan the QR code or enter the code shown in the connect window
- Both your laptop and device **HAVE** to be on the same WiFi network for this to work!

Other ways of connecting

- Use USB connection if WiFi isn't available
- Use an Emulator if you don't have an Android device
- Instructions are on the setup page:
<http://appinventor.mit.edu/explore/ai2/setup.html>

Setup is Complete!

- The WiFi (and USB) connection is a live connection where any changes in your App show up immediately on your device. If the connection is broken, then the app is gone from your device.
- We can also build the App as an Android package (apk file) from the *Build* menu and then install it on our device. Then the app is actually installed on the phone and stays on it unless you remove it.

Create your first app!

- Go to the following website to find the HelloBunny tutorial:
 - <http://coen.boisestate.edu/idocode/cc1/>
 - *HelloBunny* app
- You can also recreate the TalkToMe app.
 - Play with the speech pitch and speed rate!