App Inventor Lab
Programming in AI2

- AI2 is an example of an IDE (Integrated Development Environment)
- Everyone should be able to tinker with their mobile devices (maker v. downloader)
- Can fill the needs of a person, family, social group, business, or community
Magic 8 Ball App #3

- Accelerator Responds to Shaking Event
- Text to Speech
- List Variables
Magic 8 Ball App #3

- Bonus Challenge
  - Fix the Bug!
  - Add App Icon
  - Add additional responses
Another IDE - xCode
Swift Playgrounds

Learn serious code on your iPad.
In a seriously fun way.
Programming in AI2

- AI2 - Play, Prototype, Personal Apps, Develop Complete Apps
- AI2 deals effectively with events, options, mistakes are limited, language frustrations minimized
• Every app is constructed out a set of components chosen from the Designer

• Some components are *visible* and appear on the screen of the app (e.g. buttons, labels, text boxes, canvases, etc.)

• Some components are *non-visible* --- you can't see them on the screen, but they are needed for the app to behave the way you want (e.g., clocks, accelerometer, camera, etc.)
• Each component has a set of properties that specify how it appears and how it behaves. For example, the properties of a button include its width and height, its background color, and the text that appears on the button. The properties of a clock include its interval (how often its timer goes off) and whether its timer is currently enabled.
AI2 - Event Handlers

- All App Inventor apps are event-based, which means that they respond to events from the user (button press, shake device, etc.)

- AKA Declarations - Top-level block that cannot be composed with other blocks
Every component can have getter blocks that return the current value of a property and setter blocks that change the current value of a property.
Method Blocks perform special actions associated with those blocks. You can generate your own methods!

Methods allow you to combine multiple steps that can be reused, resulting in efficient code!
AI2 - Expressions & Statements

- **Expressions** - Denotes a *Value*
  - Button1. BackgroundColor
  - Label1. Text
  - Canvas1. PaintColor
  - 17
  - 1 + 1
  - True
  - Random integer from [ ] to [ ]

- **Statements** - Perform an *Action*
  - Set {Button1. BackgroundColor} to
  - Set {Label1. Text} to
  - Set {Canvas1. PaintColor} to
  - Call Camera1. TakePicture
  - Call TextToSpeech1. Speak message
  - Call Canvas1. DrawCircle x y r
  - Call Canvas1. DrawRectangle x y w h
  - Call Canvas1. DrawEllipse x y w h
  - Set {Canvas1. DrawColor} to
More on Statements

• Statements compose vertically, and their order matters: actions are performed from top to bottom
Putting It All Together

- An App Inventor app consists of a collection of event handlers that are filled out with other blocks indicating what's supposed to happen when the events fire (are activated).

- Explain what the following app is supposed to do (assume Label1 is initially labeled with 0):

  ```
  when Clock1 .Timer
  do set Label1 .Text to 1 + Label1 .Text
  ```

  Specifies properties of the Component

  ```
  when Button1 .Click
  do set Label1 .Text to 0
  ```

  Specifies properties of the Event

  ```
  when Ball1 .Flung
  do set Ball1 .Speed to get speed
  set Ball1 .Heading to get heading
  ```
Variables in AI2

- Defining a variable reserves a location in your device’s memory.

```code
initialize global name to "Score"
```
Blocks Editor
UNC Ram Mash #4

- Touch the Ram
  - Ram Random Move
  - a Clock to control movement
  - Vibrate when touched
UNC Ram Mash App
Extras # 4

• Add Score and Score label
• Score Goal and “You Win!” Text to Speech
• Stop Ram and end of Game
• Add Reset Function
Distributing Your App
Exporting App

- **.AIA file** - Editable and reusable source code (blocks)
- **.APK file** - Directly download and install on phone
- Can share the .AIA file to the App Inventor Gallery (or remix apps that others have shared)
- Can export the .APK file to the Google Play Store for distribution
Feedback Form