Events-First Programming in App Inventor



Lyn Turbak Wellesley College



Mark Sherman and Fred Martin
UMass Lowell



David Wolber
University of San Francisco



Shaileen Crawford Pokress
MIT Media Lab

CCSCNE, Providence College, April 25, 2014

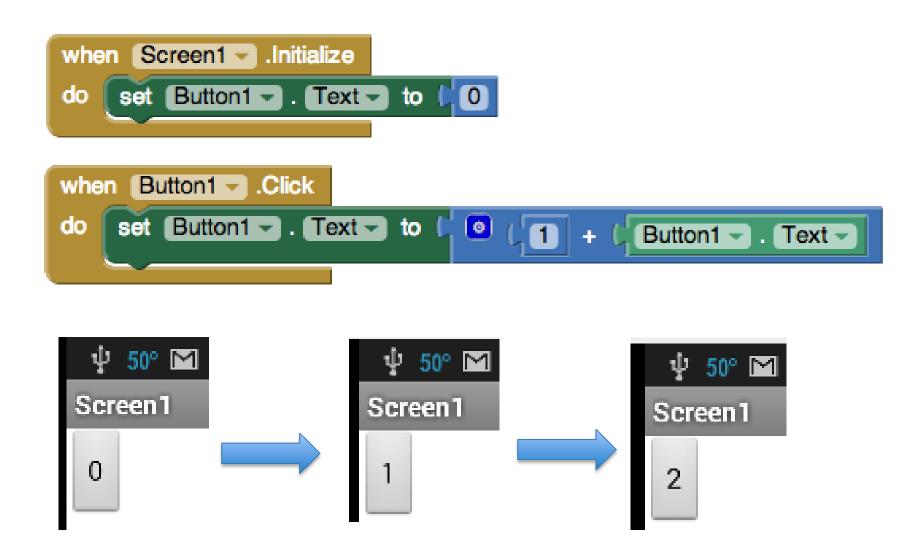
Talk Overview

- Event-Based Programming in App Inventor 2
- Surprises in the App Inventor Event Model
- Event-Based Thinking with App Inventor 2

Talk Overview

- Event-Based Programming in App Inventor 2
- Surprises in the App Inventor Event Model
- Event-Based Thinking with App Inventor 2

Simple Events: CountButton



CountButton Example in Java

```
// Assume counter, counterLabel, and incrementButton
// are instance variables
counter = 0;
countButton = new JButton("Increment"); countButton.addActionListener(
    new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        counter++;
        countButton.setText(Integer.toString(counter));
    }
}
```

Simple Events: Talk To Me

```
when Button1 .Click

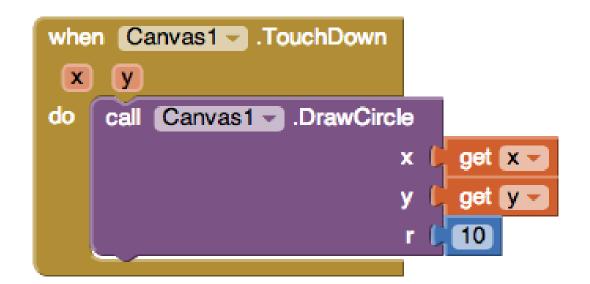
do call TextToSpeech1 .Speak

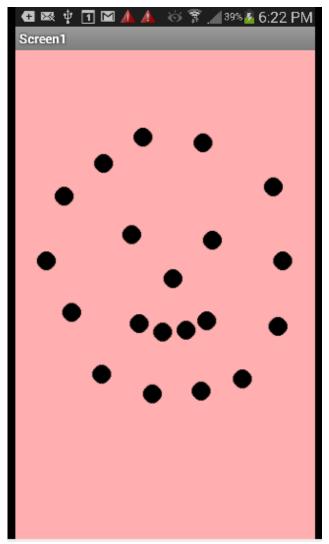
message Welcome to Providence College "
```

Simple Events: Stop Shaking Me

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

Event Parameters: Dots on Canvas





Event Parameters: No Texting While Driving

```
when TextSMS .MessageReceived
 number
        messageText
                             to get number -
do
   TextSMS - Message -
                          to I'm driving now. I'll text you later. "
      TextSMS 		■ .SendMessage
      o join
                                     New text from
                   message
                                    get number -
                                     The message says
                                    get messageText
```

Callback Events: Speech to Text

```
Button1 - .Click
when
         SpeechRecognizer1 - .GetText
when SpeechRecognizer1 .AfterGettingText
  result
        TextToSpeech1 		■ .Speak
                                    get result -
                         message
                   . Text -
                            to
                               🛴 get result 🗸
         Button1 -
     set
```

Callback Events: Load Web Page

```
LoadPageButton -
                         .Click
when
do
                                  http://appinventor.mit.edu
         Web1 🕶
                   [ Url⊸
                           to
          Web1 ▼
                  .Get
      Web1 	☐ .GotText
when
                                        responseContent
 url
       responseCode
                        responseType
    set TextBox1 - . Text - to
                                     get responseContent -
do
```

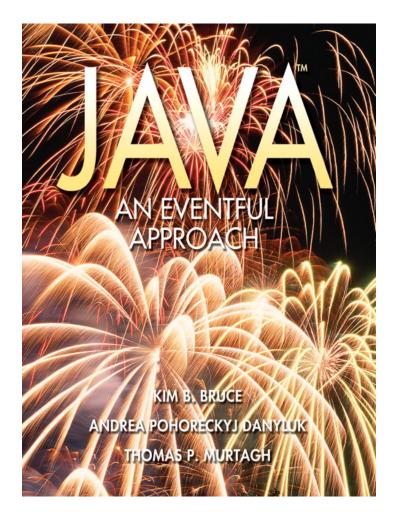
Web Example in JQuery JavaScript

```
$.get("http://appinventor.mit.edu",
  function (page) { alert(page); }
);
```

Events First!

- Interactions with mobile apps are inherently event based.
- Want an event-based programming model that matches the event-based user model.
- Accessible to novices.
- Powerful paradigm for professionals.
- App Inventor programs are just a collection of event handlers.

Java: An Eventful Approach



- "We feel it is clearly more appropriate to adopt programming tools that reflect natural human models of the task undertaken than to adjust the thinking patterns of our student to fit the limitations of the programming techniques we present."
- Java objectdraw library allowed eventbased programming from week 1.
- Including event-based techniques enhanced (and did not displace) other topics.
- Students thrilled by real-world interactivity of EBP.

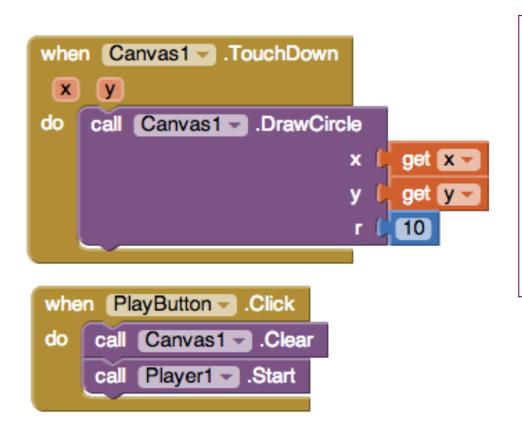
Talk Overview

- Event-Based Programming in App Inventor 2
- Surprises in the App Inventor Event Model
- Event-Based Thinking with App Inventor 2

Events in the App Inventor Notional Machine

- Only one event handler can be executing at any given time.
- Other events that occur while an event handler is executing are queued and handled later, in order.
- Any GUI changes during an event handler are not displayed until the event has completed.
- Certain system actions (playing a sound file, initiating a web request, etc.) are executed in a thread separate from the current event handler.
- Playing a sound on a Player component first terminates any sound currently playing except when the source file has not been reset, in which case the new play request is ignored if the sound is already playing.

Can dots be drawn while the sound plays?



Answer: **YES**.
The sound plays in a different thread, so the PlayButton.Click event completes quickly, allowing Canvas events to run.

Can dots be drawn while the sound plays?



Can we see the counter increment while the sound plays?

```
when PlayButton .Click

do call Canvas1 .Clear

call Player1 .Start

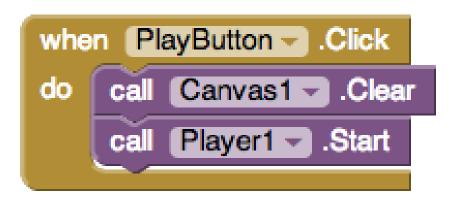
set Counter . Text to 0

while test Player1 . IsPlaying .

do set Counter . Text to 1 + Counter . Text .
```

Answer: **NO**. The GUI is not updated until the event handler finishes executing, when the counter reaches its final value.

Suppose PlayButton is clicked to play a sound. What happens if it's clicked again while the first sound is playing?



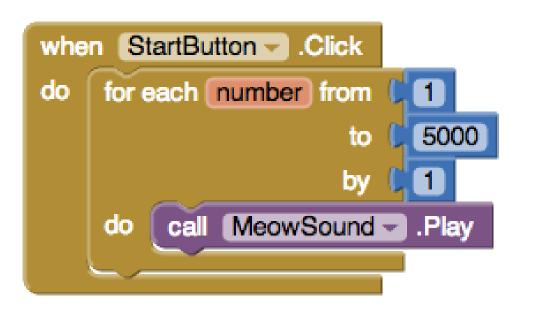
Answer: **NOTHING**. As before, the first PlayButton.Click event completes quickly. When the second event executes, the second attempt to play a sound is ignored because the same one is already playing.

What happens when PlayAllButton is clicked?

```
PlayAllButton . Click
    for each index from [1]
do
                           length of list list | get global recordings -
              PlayAllButton ▼ . Text ▼ to leget index ▼
    do
              Player1 - . Source - to select list item list get global recordings -
                                                       index
                                                               get index -
              Player1 - .Start
        PlayAllButton - . Text -
                                  to ( Play All "
```

Answer: Only the last recording is played because each Player1. Start terminates the previous recording. Also, the button text does not display any of the intermediate indices.

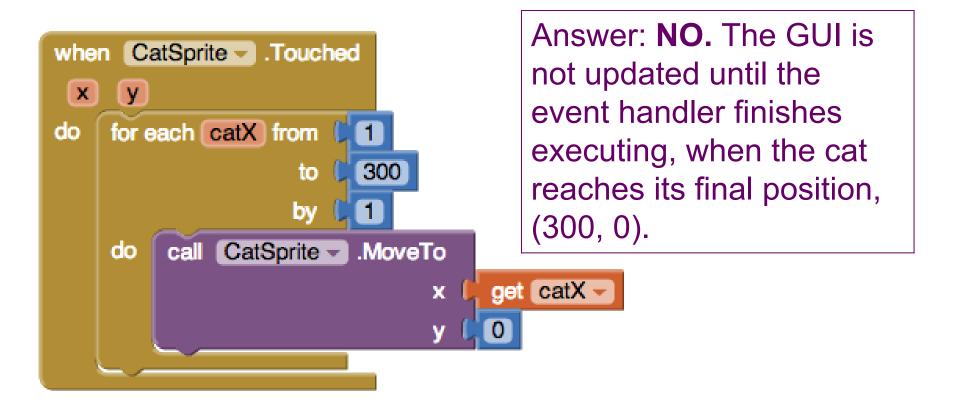
How many times is MeowSound played?



Answer: Hard to predict, but way less than 5000 times. The for loop executes quickly, and most requests to play MeowSound are ignored because the same sound is already playing.

In practice, the answer is about a dozen times.

Do we see the cat walk across the screen?



What happens if we click Start and touch the cat?

```
when StartButton .Click

do for each number from 1

to 5000

by 1

do call MeowSound .Play
```

```
when CatSprite .Touched

x y

do for each catX from 1

to 300

by 1

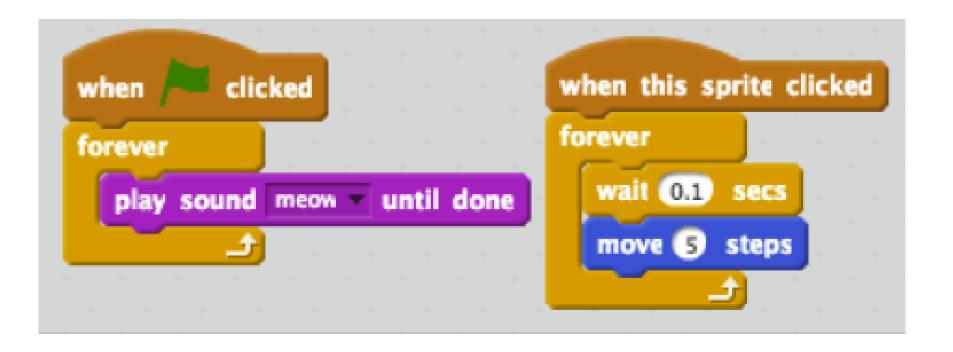
do call CatSprite .MoveTo

x get catX y

y 0
```

Answer: MeowSound is played about a dozen times, during which the cat touch event is queued. After the sound playing completes, the cat jumps to position (300, 0).

Event Handler Interleaving in Scratch



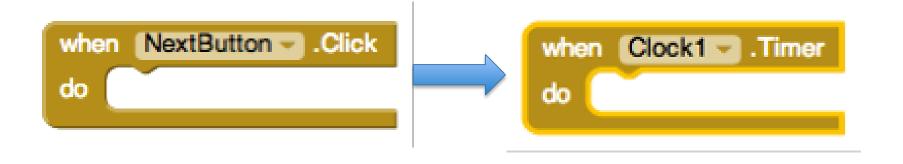
Talk Overview

- Event-Based Programming in App Inventor 2
- Surprises in the App Inventor Event Model
- Event-Based Thinking with App Inventor 2

Global State Machines: Manually Advancing Slideshow

```
initialize global index to
                                initialize global pictures to
                                                          make a list
                                                                          cat.jpg
                                                                           dog.jpg
when Screen1 - Initialize
                                                                           rabbit.jpg
do
    set global index - to 11
                               select list item list 📜 get global pictures 🕶
        index (
     NextButton - .Click
do
               get global index -
                                 er length of list list get global pictures
          set global index - to 11
    then
    else
          get (global index 🗸
                               select list item list get global pictures -
        Slide 🕶
                 Picture - to
                                                  get global index -
```

Timers: Automatically Advancing Slideshow



Working Player for Recording, Part 1

```
initialize global index to
      PlayAllButton . Click
    set global index - to
    call playNextRecording
     Player1 . Completed
    set global index - to
                             (0)
                                 🕻 get (global index 🗸
    call playNextRecording -
```

Working Player for Recording, Part 2

```
to playNextRecording

do if get global index > length of list list get global recordings > length of list list
```

Scratch Example Revisited

```
when clicked

forever

play sound meow until done

play sound meow until done

move 5 steps
```

```
when StartButton .Click
do call Player1 .Start

when Player1 .Completed
do call Player1 .Start
```

```
when CatSprite . Touched

x y

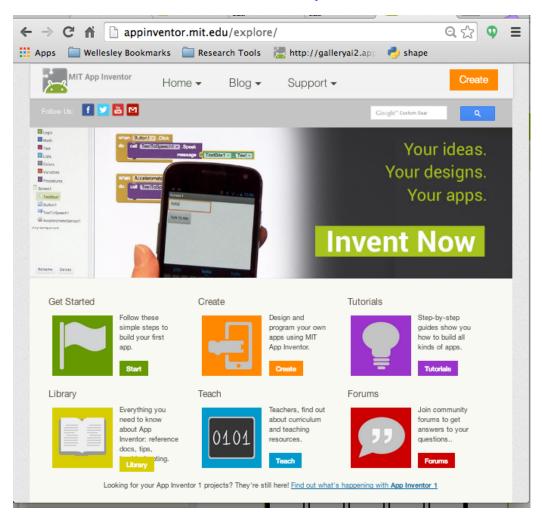
do set Clock1 . TimerInterval to 100

set Clock1 . TimerEnabled to true

when Clock1 . Timer

do set CatSprite . X to CatSprite . X + 5
```

Thank You! Questions?





Acknowledgment: This work was supported by the National Science Foundation under Grants 1225680, 1225719, 1225745, 1225976, and 1226216.

3-Day App Inventor Workshop for Undergraduate Faculty





Computational Thinking through Mobile Computing





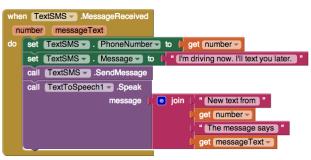
Mon July 21 – Wed July 23, 2014 University of Massachusetts Lowell

Workshop highlights:

- Hands-on sessions learning MIT App Inventor 2, with new browser-based blocks programming interface
- Teach computational thinking in CS0/CS1 while your students build Android phone & tablet apps
- Whole-course designs and modules for existing courses for majors and nonmajors
- Resources: videos, screencasts, tutorials, quiz-making environment, sample student projects
- Rubric for assessing mobile computational thinking
- Field trip to MIT Media Lab

Workshop Leaders: Learn from the creator of App Inventor and some of the early teaching pioneers.

- Hal Abelson (creator), Shay Pokress, Josh Sheldon (MIT)
- Ralph Morelli (Trinity College)
- Dave Wolber (University of San Francisco)
- Fred Martin, Karen Roehr, Mark Sherman (UMass Lowell)
- Eni Mustafaraj, Franklyn Turbak (Wellesley College)
- Larry Baldwin (BIRC, project evaluator)



More details and application:

http://bit.ly/mobileCT-workshop-2014

Preference given to those who plan to use App Inventor in their courses

\$500 stipend. Apply now!

Application deadline: Apr 28, 2014.



http://bit.ly/mobileCT-workshop-2014