

MoMu: A Mobile Music Toolkit

Nicholas J. Bryan, Jorge Herrera, Jieun Oh, Ge Wang

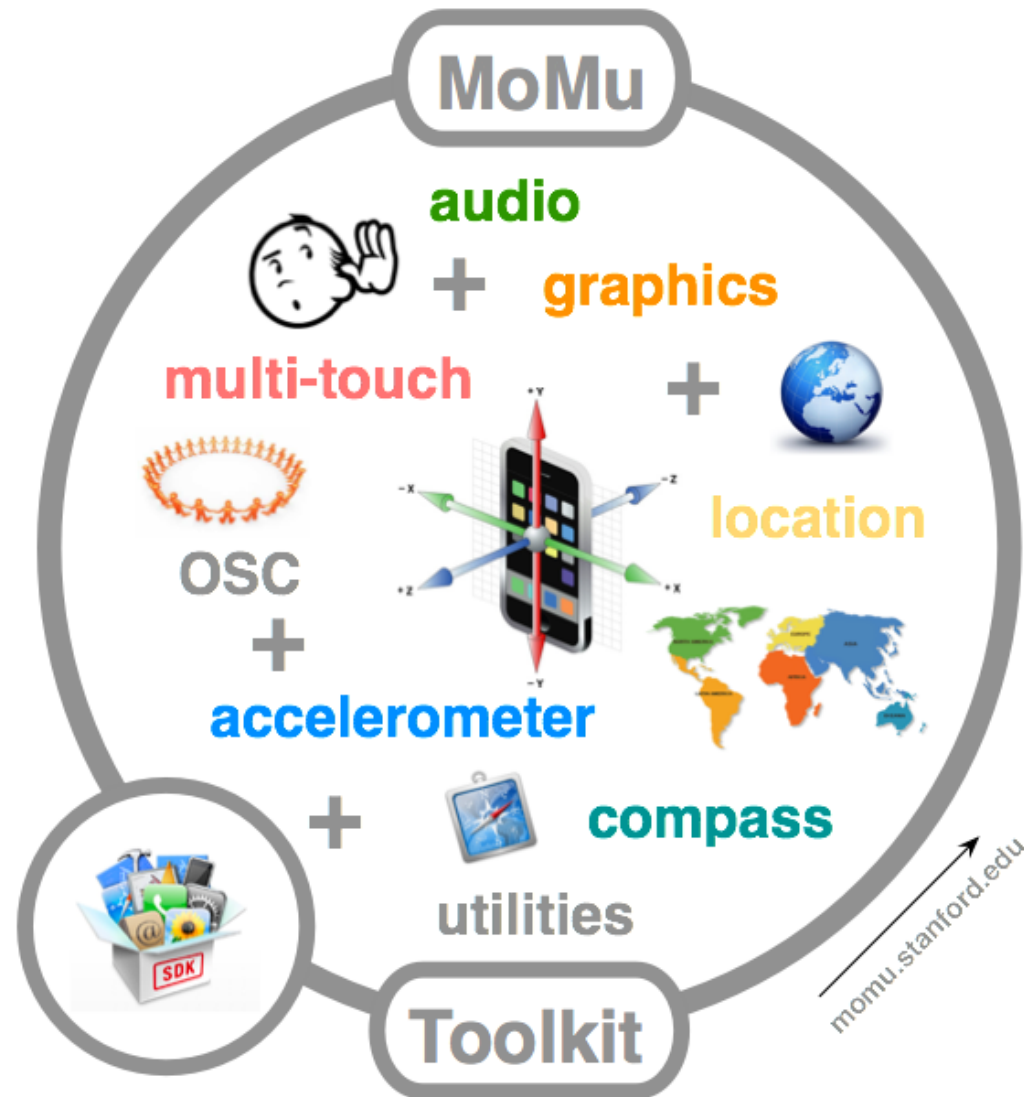
Stanford University | CCRMA



NIME 2010

<http://momu.stanford.edu/>

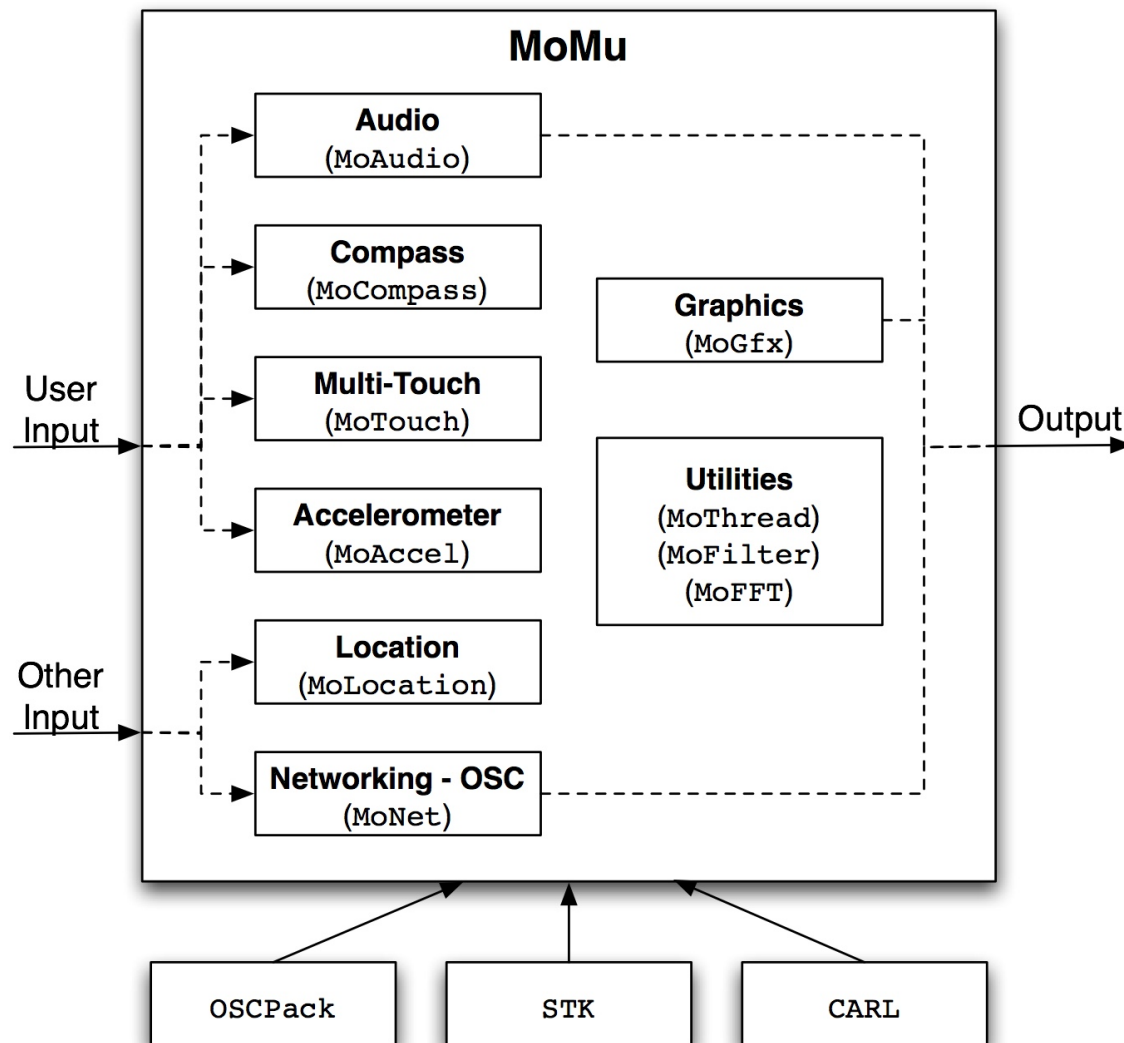
Interactive Music Applications for Mobile Devices



Design Goals

- Real-time audio, synthesis, and control
- Consistent conventions for external sensor access
- Unified common functionality for mobile music
- Focus on ease of use, setup, and installation
- Open source C, C++, and Objective-C code

iPhone Developer



iPhone SDK

Onboard Sensors

Audio
(MoAudio)

Compass
(MoCompass)

Multi-Touch
(MoTouch)

Accelerometer
(MoAccel)

Location
(MoLocation)

Networking - OSC
(MoNet)

- Resource server for each sensor
 - Polling
 - Interrupt-driven event handling
 - When applicable, both!
- Implementation
 - Static C++ classes
 - Multiple callbacks
 - Pass data through using *void ** function parameter
- Typical Code

```
// Setup
```

```
MoSensor::addCallback( ... );
```

```
// Interrupt-driven handling
```

```
void moSensorCallback( ... )  
{ ... }
```

Audio I/O: MoAudio

Setup

```
// Initialized the real-time audio  
MoAudio::init( sampleRate, bufferSize, numChannels );  
  
// Start the audio callback  
MoAudio::start( audioCallback, NULL );
```

Sensor Access

```
// Callback implementation  
void audioCallback(Float32 * buffer, Uint32 bufferSize, void *  
userData)  
{  
    for( int i = 0; i < numFrames; i++)  
    {  
        buffer[2*i] = buffer[2*i + 1] = synthesis.tick();  
    }  
}
```

Compass: MoCompass

Setup

```
// Initialized compass  
MoCompass::addCallback( compassCallback, myDataStruct );
```

Sensor Access

```
// Callback implementation  
void compassCallback( CLHeading * heading, void * userData )  
{  
    // Get the user data  
    MyDataStruct * data = (MyDataStruct *) userData;  
  
    // Process  
    ...  
}
```

Accelerometer: MoAccel

Setup

```
// Initialized accelerometer  
MoAccel::addCallback( accelCallback, NULL );
```

Sensor Access

```
// Callback implementation  
void accelCallback(double x, double y, double z, void * userData)  
{  
    // Process  
    ...  
}
```

Location: MoLocation

Setup

```
// Initialized location  
MoLocation::addCallback( locationCallback, NULL );
```

Sensor Access

```
// Callback implementation  
void locationCallback( CLLocation * newLoc, CLLocation * oldLoc, void * data )  
{  
    // Process  
    ...  
}
```

MultiTouch: MoTouch

Setup

```
// Initialized multiTouch  
MoTouch::addCallback( touchCallback, NULL );
```

Sensor Access

```
// Callback implementation  
void touchCallback( NSSet * touchSet, UIView * view, const  
std::vector<UITouch*> & touchVec)  
{  
    // Process  
    ...  
}
```

Networking (OSC): MoNet

Setup

```
// Initialize OSC listener  
MoNet::addAddressCallback( osc_address, oscCallback, NULL );  
MoNet::setListeningPort( port );  
MoNet::startListening();
```

Access Incoming message

```
// Callback implementation  
void oscCallback( osc::ReceivedMessageArgumentStream & oscin,  
                 void * data )  
{  
    // Process  
}
```

Send message (no setup required)

```
// Send OSC message  
MoNet::sendMessage( std::string server, int port,  
                   std::string osc_address,  
                   char types[], int num_vars, [vars] );
```

Setup Evaluation

- *Real-time audio (2 line setup)*
- *Accelerometer (1 line setup)*
- *Compass (1 line setup)*
- *Location (1 line setup)*
- *MultiTouch (1 line setup)*
- *OSC Networking (0/1 send/receive line setup)*

Other Useful Abstractions

Graphics
(MoGfx)

Utilities
(MoThread)
(MoFilter)
(MoFFT)

- Audio Synthesis/Processing with the Synthesis Toolkit (STK)
- Common graphics functionality not available in OpenGL ES
- Threading
- General Filtering
- Fast Fourier Transform (FFT)

Sound Synthesis/Processing

- MoMu STK
 - <http://momu.stanford.edu/stk>
 - Synthesis Toolkit in C++ (STK) v. 4.4.2
 - Perry Cook & Gary P. Scavone
 - <http://ccrma.stanford.edu/software/stk/index.html>
- MoFFT
 - UCSD CARL distribution of music software

Beta Evaluation

- Stanford Mobile Phone Orchestra (Mopho)
 - Numerous concerts throughout the past year
 - Concert this Friday with instruments using MoMu!
- Mobile music development and design class
 - Used as part of a 9 week long course (Stanford, CA)
 - Week 1: full-duplex audio, accelerometer control, multi-touch, user interface elements

Conclusions

- Interactive mobile music applications
- Focus on ease of use, setup, and installation
- Consistent external sensor access
- Numerous other useful programming tools for mobile music
- Open source C, C++, and Objective-C code
- BSD-like license

Acknowledgments & Thanks!

- National Science Foundation Creative IT grant No. IIS-0855758
- Spencer Salazar of Smule
- Luke Dahl, Rob Hamilton, Turner Kirk
- Winter 2010 mobile music class at Stanford University
- Multiple authors of prior work incorporated into MoMu (STK, OSCPack, CARL)



MoMu: A Mobile Music Toolkit

<http://momu.stanford.edu/>

