immersion.

Bob Heubel
Developer Evangelist, Haptics

November 2013

NASDAQ: IMMR

Enhancing the mobile user experience through good tactile design

Who is Immersion? Company Highlights

Immersion: The Haptics Company

NASDAQ: IMMR

Established in Large, Rapidly Growing Markets

Technology shipped in over 1B devices

- Mobile phones, game controllers, auto, industrial, casino, other portable devices

Technology Leader

Developing tactile touch solutions also known as "Haptics" for over 20 years.

World Class Customer Base

Samsung, Nokia, LG, Toshiba, Fujitsu, Sony, Microsoft, Logitech, Lexus, BMW, CAE and many more

Strong Intellectual Property

Portfolio of 1,300+ granted and pending patents specifically in the field of Haptics related to both hardware and software

Worldwide Support

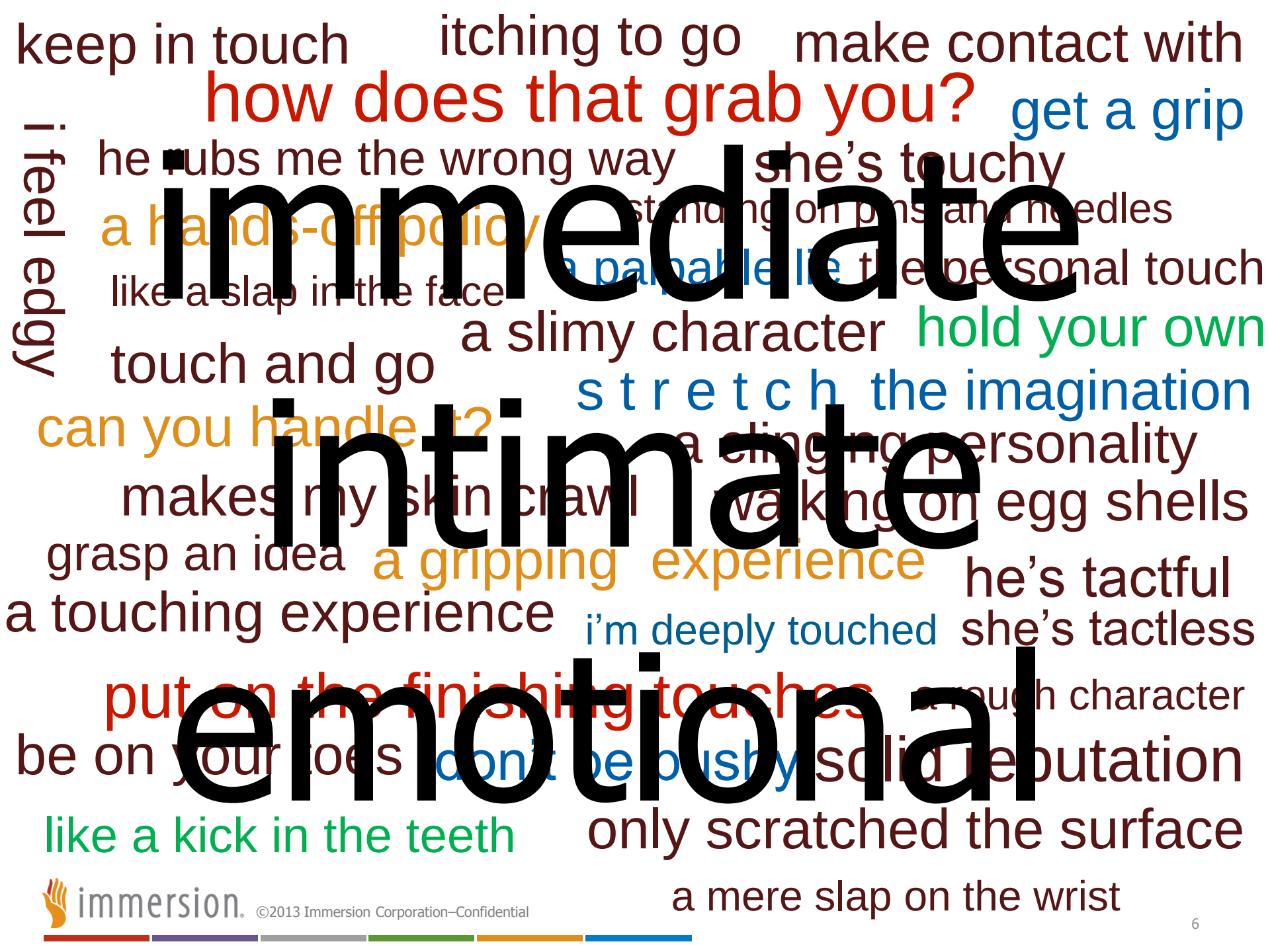
Offices in EU, Korea, Taiwan, Japan, US & Canada

Atlanta
75
Customer Service
Regarding your order #9634029
Christina Brenton
Final Presentation with Notes
Facebook
You have 2 friends with birthdays
Terri Nowak
LG
Camera
Gallery
3D
SAMSUNG
12:25
PM
Mon, September 05
85°F
Dallas
Clear
WED
THU
FRI
SAT
SUN
2:45 PM
OUR HAPPY MOMENT.
Phone
Contacts
S Memo
Messaging
Applications
12°C
Barcelona
Cold with partial sunshine
13 Sunday
7:00-8:00 PM
Watch a Live Broadcast of the GALAXY S II Unpacked Event
Contacts
Camera
Gallery
Samsung Apps
Market
AllShare
Maps
Talk
Gmail
ThinkFree Office
Qik Video
Skype
Music
Video
Internet
10:20 AM
Modern Rock
ShoreTel
12 : 34
TSUTAYA TV

Say hello to Homunculus

The Mobile User


Point 1: Our sense of touch works with our embodied cognition to affect the user experience.



Point 2: Touch feedback fulfill a need for tactile gratification that is sorely missed in media.

Social Isolation

Point 3: Tactile feedback makes human communication more intimate and emotional.

Design goals when using tactile feedback

- Make mobile devices feel more usable
- Make apps feel more satisfying
- Make communication with friends and family feel more personal

Core tactile feedback design principles

- Simple sensations are often the most effective
- Sensations that fit with visual & audio elements make the whole greater than the sum of its parts
- It is bad to annoy, confuse or overwhelm the user
- It is good to give the user options

Always "play" test your tactile effects to ensure that you are meeting these core principles.

Lessons learned from console games

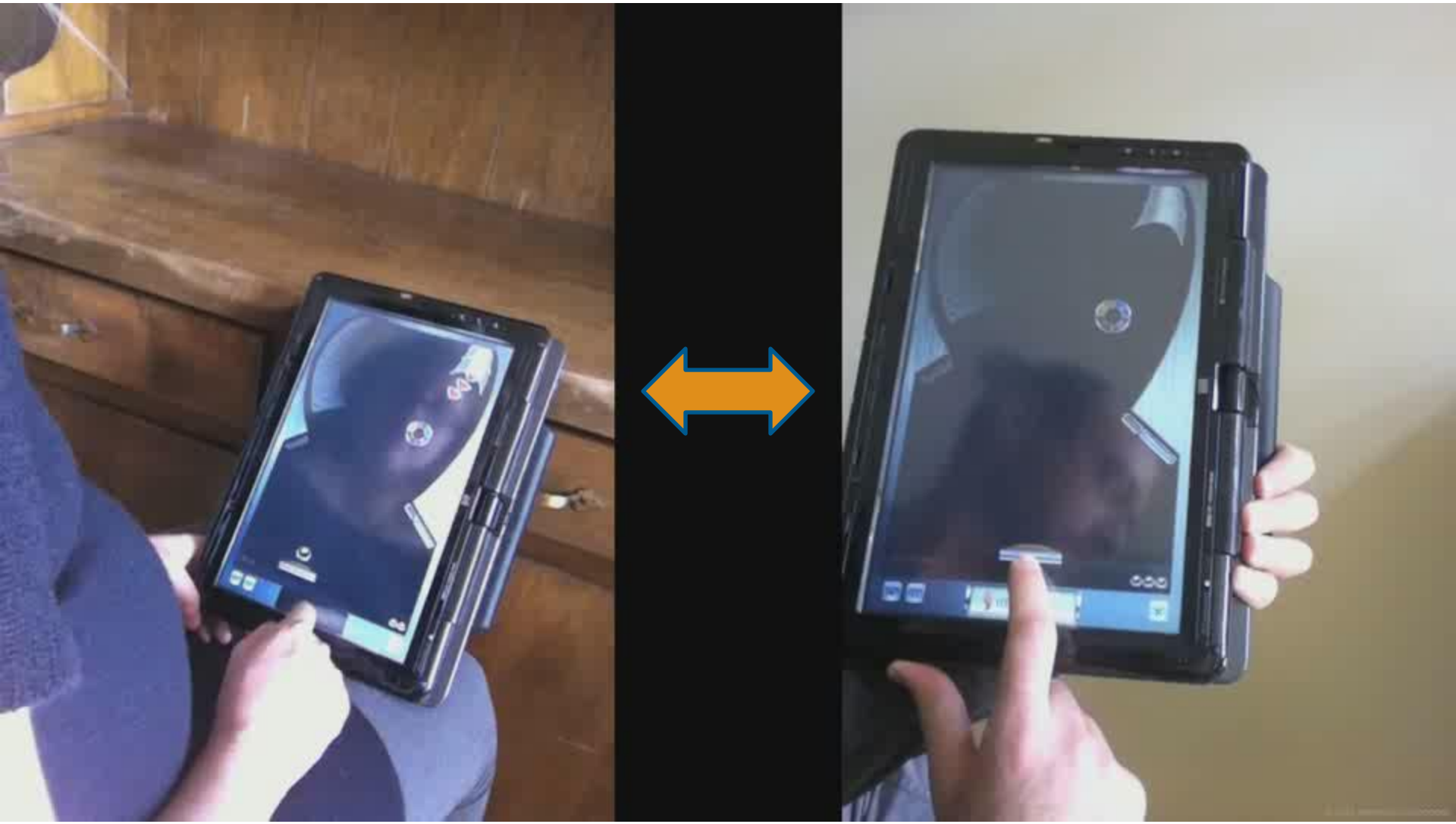
All major gaming consoles use tactile effects to help immerse gamers in their virtual worlds.

PlayStation - Gran Turismo

X-Box
Halo

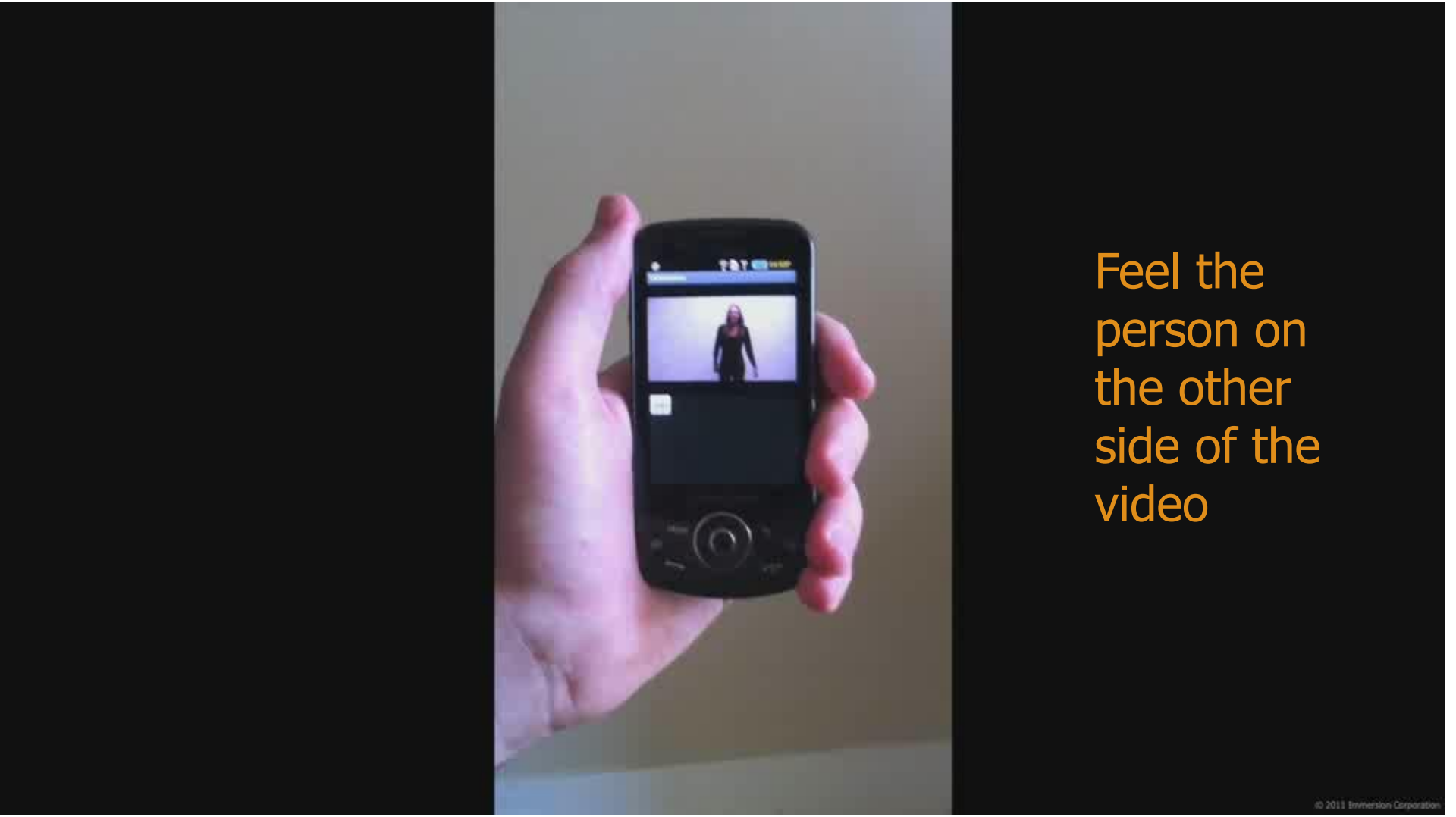
Nintendo Wii
Legend of Zelda

Interactive tactile game example



Two-way interactive touch example

Tactile video example

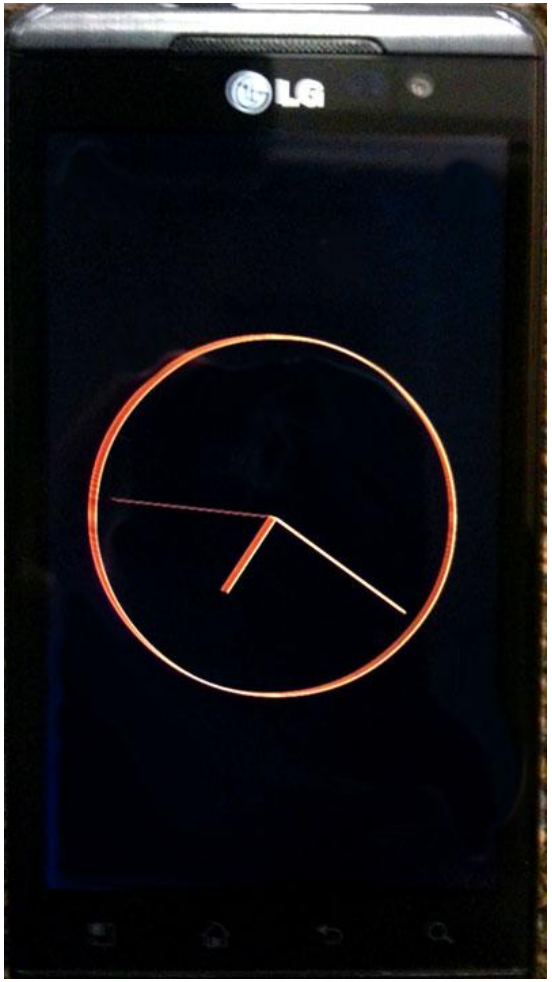


Tactile feedback effects beyond games

Feedback in games is natural like console gaming rumble, but consider other possible uses like…

- Screen gestures – swipes, slides, pinches, twists
- Social networking – virtual poking, winking, smiling, kissing
- Children's learning – touch confirmations & answer rewards
- Interactive 2-way videos – shared screen virtual touching
- Customizable alerts – different alert patterns for caller & messaging IDs
- eReaders – feeling page turning, writing notes
- Apps for the sight impaired – navi-cues, Braille input, tactile watch
- Sports / Health Apps – alerts based on health monitoring or pacing

Tactile feedback clock for everyone



Tactile Clock

Allows you to feel the time.

Good for both blind and sighted users.

Top 5 reasons to use tactile feedback effects

#5 Increased sense of realism

"Feels like the real world"

#4 Increased immersion combining audio/visual/touch

"Sum is greater than individual parts"

#3 Greater user satisfaction

"Proven in game platforms"

#2 Increased stickiness

"Greater emotional connection to game play and UI"

And the #1 reason to use tactile effects…

#1 The potential for increased downloads

"Differentiating your app"

Grand Theft Auto III images courtesy of Rockstar Games.

How to program tactile effects into mobile apps

Tactile effect design by operating system

Nearly all operating systems allow you program vibration events into your applications. But not all methods are created equal.

iOS – Only allows you to call a vibration constant that triggers the motor for a set duration. This is not good for most games.

Android – Allows you to call a *Vibrator* Class and set on/off duration times. Also offers an extended method through Immersion API.

Tizen– Allows you native and web app programming environments. Both use a *Vibrator* Class but native allows for intensity control. Whereas web has no intensity control and uses a *navigator.vibrate* method. Both have methods for pattern arrays.

Blackberry – Allows you the same control as Android but uses a *VibrationController* Class. Also has a *Intensity* parameter

Windows – Allows you basically the same control as Android & Blackberry but uses a *VibrateController* Class, but without intensity control.

Symbian – For Javascript you use start/stop methods with their *Vibra* Class that allows you to set duration and intensity parameters. For Qt on their S60 API you use a wrapper class called *HapticFeedback*. This Additionally, the HapticFeedback Class can be extended method through the Immersion API

Hurdles to good mobile tactile design

- Time consuming
  - Manual programming of sensations, usually setting durations and pulsing patterns in milliseconds and intensity levels on a scale of 1-100 (if available)
- Inconsistent feel
  - Operating systems use different vibe classes and methods
  - Parameter controls vary between operating systems
  - Motor types vary from manufacturer to manufacturer
- Differing user preferences
  - Touch feel is personal… varying by age, sex and region

How to jump the tactile design hurdles

- Saving tactile design time
  - Reuse your effect parameters from project to project
- Getting a consistent feel
  - Design and test on the highest volume devices
  - Create manufacturer specific themes with parameters optimized for each target manufacturer
  - If you can't get a good feeling from a specific device, exclude it from your design
- Compensating for user tastes
  - Set smart defaults but allow options in settings for users to increase/decrease the intensity of your tactile effects to their personal tastes, the same as audio

Lowered hurdles specifically for Android

Because the Android operating system is the most open for tactile design and holds the lion share of the smart phone market…

Immersion created a **free** pre-designed library of tactile effects that compensates for vibration motor differences so developers can quickly add tactile sensations into their apps and games.

We call this the…

"Immersion Haptic Development Platform for Android"

And we hope to offer additional support for this same library in other operating systems in the future.

Immersion Android apps for tactile design

Two FREE Android apps available on Google Play for developers to preview the library of effects and view tactile gaming scenarios.

Haptic Muse

(effects in gaming scenarios)

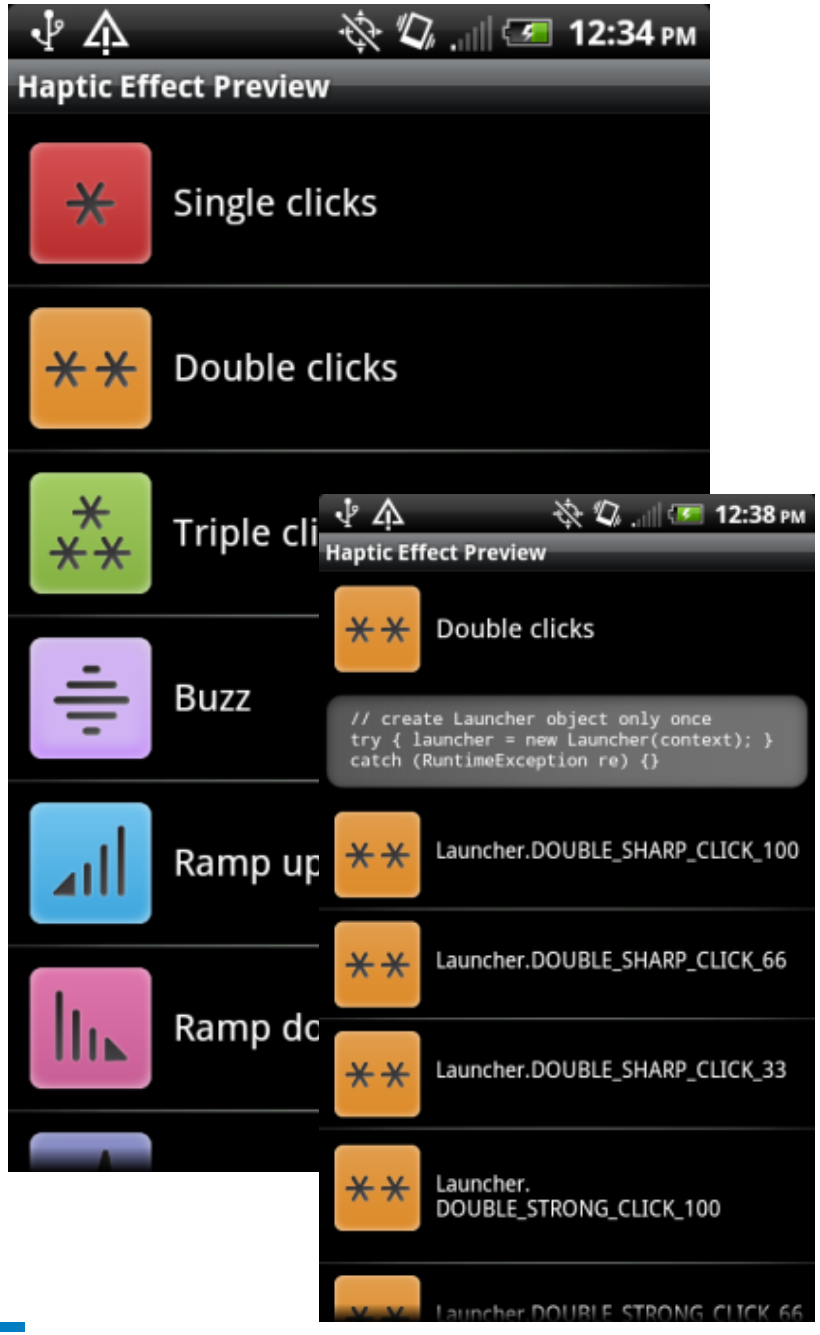
&

Haptic Effect Preview

(effects in category list form)

Haptic Effect Preview App

- 124 pre-designed tactile effects
- Free app on Google Play
- Feel each tactile effect on any Android device before programming
- Code sample provided for each effect



Immersion Haptic Muse App

- Seven different gaming effect zones
- See, hear and feel typical game effect scenarios
- Includes sample code identifying each effect in the use cases

Benefits of the extended method for Android

Google Vibrate	Immersion Extended Method
Inferior battery usage: only controls the duration of vibration	Very efficient use of battery: control over duration, magnitude and frequency of vibration
1 basic effect – full magnitude w/durations set manually	Library of 124 pre-made gaming and user interface effects
Ineffective for longer duration game effects, only gives you buzz	Ideal for longer duration subtle game effects
Unreliable feel across Android devices due to different motor types	Most consistent feel across **ALL** Android devices

“Consistent feel across ALL Android devices”

Why is this important?

Because there are over 600 Android device makers and over 7000 models of handsets and tablets that all use different vibration actuators like these:

- 3 Distinct Generations:

- Eccentric Rotating Mass
- Linear Resonant Actuator
- Piezo Electric

Haptic SDK Quick Start Guide

GET STARTED

1 Download Resources

- Immersion's Haptic SDK
  www.immersion.com/haptic/sdk
- Immersion's Haptic Effect Preview App
  www.immersion.com/haptic/previewApp

2 Add UHL to Eclipse Project

- Extract UHL_x_x_xx.zip
- Copy extracted **libImmEmulatorJ.so** file to your Eclipse project **libs/armeabi** folder (create folders if necessary)
- In Eclipse, refresh project to see **libs/armeabi/libImmEmulatorJ.so**
- For ADT 16 or earlier, navigate to Project > **Properties** > **Build Path** > **Configure Build Path**, Click on **Add External JARs**..., browse to the location of the extracted **UHL.jar** file
- For ADT 17 or later, copy **UHL.jar** into your Eclipse project **libs** folder
- In the **Permissions** tab under **AndroidManifest.xml**, add "**android.permission.VIBRATE**" for all build configurations

3 Add Import Statement

- Import the Launcher class wherever the Launcher will be used

```
import com.immersion.uhl.Launcher;
```

4 Add Launcher Member

- Add a Launcher member to the main Activity class or other application-wide class

```
private Launcher m_launcher;
```

5 Instantiate Launcher

- Instantiate the Launcher object once, usually in the main Activity onCreate function

```
try
{
    m_launcher = new Launcher(this);
}
catch (RuntimeException e)
{
    Log.e("My App", e.getMessage());
}
```

For detailed information about any of these steps, visit: www.immersion.com/haptic/guide

6 Launch Haptic Effect Preview App

- Use the Haptic Effect Preview application from the Android Market to feel the built-in effects and determine which effect IDs to use in your application.

7 Play Haptic Effect

- Play a haptic effect

```
try
{
    m_launcher.play(Launcher.BOUNCE_33);
}
catch (RuntimeException e) {}
```

Use Haptic Effect Preview Application on your target Android phone to choose effect.

8 Stop Haptic Effect

- Stop a haptic effect

```
try
{
    m_launcher.stop();
}
catch (RuntimeException e) {}
```

Add this to Activity onPause function in case a phone call is received.

Additional tactile design resources

Immersion's Haptic SDK Tools:

Including plugins for Unity3D, Marmalade, Game Maker

www.immersion.com/haptic/sdk

Enough Software's

Mobile Developer Guide to the Galaxy:

www.enough.de/products/mobile-developers-guide/

Wireless Industry Partner's (WIP) Design Guide

Mobile Developer's Guide to the 5th Dimension:

wip.org/download/**Fifth_Dimension**_v1.pdf

Thank you!

- Contact Us HapticsDev@immersion.com
- Like Us http://www.facebook.com/ImmersionDeveloper
- Follow Us @HapticsDev
- Read Our Blog http://blog.immersion.com
- Direct Access: Bob Heubel
  RHEUBEL@IMMERSION.COM