Practical UI Guidelines for Tizen Wearable Apps (Gear 2)

Victor Dibia
City University of Hong Kong.
What We’ll Talk About

• Introduction – 3 aspects of design
• Designing around Fashion/Social Norms
• Designing around Sensing Capabilities
• Designing around Form/Function.
• Common Mistakes and Tips
• Useful Libraries
• Q/A!
About Me

I enjoy building apps for mobile devices (Android, Blackberry, Windows Phone), and the Web (HTML, JS, PHP, JSP, ASP.Net).

- Victor Dibia
  @vykthur, dibia.victor@my.cityu.edu.hk
- Researcher at City University of Hong Kong
- Lead Developer, Denvycom
Some Apps I’ve Built

Foqus

Gear Ship

Blocks Gear

Dansa

Gear Tennis

Raindrops

Proudly USA
Introduction
Why Design?

50% of customers stop using their devices after 6 months
- CCS Consulting 2014

“Coming up with the flow and intuitive universal gesture required was tough”

“Few examples”
“small dimension makes design difficult.”

- 2014 Gear 2 Developer Survey.
But there is hope …

We design to maximize the strengths and minimize the weaknesses of these devices.

**Strengths**
- Sensors
- Consistency
- Social Aspects

**Weakness**
- Interface (screensize)
- Power (battery, processing)
- Storage
3 Important Aspects.

- As a software developer, (Android Java, iOS Objective C, HTML, JavaScript, Php), there are 3 important aspects of wearable app design. We can design around ..

icons: Kenneth Von Alt, Sherrinford, Noun Project
Social / Fashion Norms

• Smartwatches ARE a fashion accessory.
• Design watchfaces that are meaningful, expressive and elegant
Designing around Sensing Capabilities
Sensing Capabilities

Use multiple sensors for improved **input**.

- **Use Accelerometer Gestures**
  E.g close a notification, game control, (devicemotion api, direction api)

- **Use the new sensors** – pressure, light, UV, GPS to estimate user state.

  Eg. Instead of *asking* the user about calories, exercise reps, sun exposure, comfort etc, use sensors to *estimate*.

- **Voice and TTS**
Sensing Capabilities

Use multiple sensors for improved feedback.

- Vibrations are great for personalized feedback. Be creative. Use vibration duration to communicate with users even without glancing.

Eg. 3 vibrations can mean a meeting with the boss, or some app state has changed.
Sensing Capabilities

Try out more advanced touch gestures..
Tap, double tap, rotate, swipe (up, left, right, down), pan, pinch, zoom, to expand interaction model.

- Use the common gestures first, go from the known to the unknown.
- Remember to teach the user (add instructions).
Designing around Form and Function
Form and Function

Be Legible and Clear

- Legible text. 30px-35px font size.
- Single action per (small) screen. Aim for two buttons max per screen. Beware of small text and buttons in games.
- The Gear is meant to be glanceable. Avoid use cases that require extended continuous use (e.g. some games)
- Optimize for Space
  Remember to minify your files and remove unused scripts. Limited device storage.
Example

Vs.

Disaggregate – One main function per screen.
Common Mistakes and Tips
The Emulator is ...

NOT always your friend. Beware of Legibility and display discrepancies between the Emulator and a Real Device

Nice game but for those with good eyes.

- User comment.
Emulator Issues.

Beware of small fonts and legibility issues. Major pain point for users.
Test/Debug faster with the Chrome Browser

- Open File with Chrome
- F12 > Escape > Emulation
- Configure Screen Size
Use Relative Dimensions for Multidevice Support

Primarily use % or auto instead of px values for CSS element positioning.

```css
margin: auto;
margin: 50px 40px 40px 100px;
```
Localize Your App.

Translate to multiple languages

- Use the Wearable IDE localization wizard.
- Allows you reach more people..
- Improves your number of downloads
Certification Tips

- Use `tizen.time.getCurrentDateTime()` instead of `Date()`.
- Manage the screen display when using non-touch gestures. E.g keep the screen on during voice dictation or gesture input.
- Ensure you save your package id for future app updates.
- Use the Samsung Developer Forum for Q/A.
Some Useful Libraries
Useful Libraries

• **Hammer.js**
  Expand your touch UI Interaction Model – Tap, DoubleTap, Pan, Pinch, Rotate, Swipe

• **Pixi.js, Cocos2D-JS**
  2D WebGL renderer with canvas fallback

• **Sketch.js**
  Javascript Particle engine

• **Charts.js**
  Graphs and Charts
Conclusion

In addition to the design guidelines, here are 3 important questions ..

• What is the main function of my app?
• How does my app perform this function better than a mobile phone? (can my app be prescribed as a solution?)
• Does my app take advantage of context to provide more value? (gym, during meditation, everywhere?)
Resources

• Samsung Developer HomePage
  Download SDK, Official UI Guidelines
  http://developer.samsung.com/

• Samsung Developer Forum – Q/A

• Tutorials on Denvycom.
  Installation, database access, sensor data access, certification tips, gestures etc
  http://denvycom.com/blog/tag/gear-2/
Thank you!
Questions ?