Developing Next Generation Human Machine Interfaces (HMI)
Gavin Kistner, Product Designer
Overview

- Realizing Artistic Vision
- UI Composer
- Use Cases
  - Goals (Not Straw Men)
  - Oohs and Aahs
  - Tear it to Shreds
  - Examine the Implementation
- QA (time permitting)
Realizing Artistic Vision

- **WYSIWYG Tools**
  - Standard interactions
  - ...with room to grow

- **Rapid, Artist-Driven Iteration**
  - Experimentation is encouraged

- **Mix in a little bit of Technical Artist Geekery**
  - Made artist-friendly

- **Engineering Stuff**
  - Runs on Target Hardware
  - Fast Enough
  - Robust and Easily Debuggable
UI Composer

NVIDIA’s HMI Platform

- **UIC Studio**
  - Professional design environment

- **UIC Architect (NEW)**
  - Integrated engineering environment

- **UIC Viewer**
  - High performance runtime
UI Composer Studio

Created for designers
- Full 3D environment
- Advanced special effects
- Real-time feedback
- DCC Integration

Modern and Powerful
- Open project structure
- Multi-designer workflow
- Professional content pipeline

Slides
Modular organization of large projects — like PowerPoint “Master” Slides

3D View
Live WYSIWYG, real-time view of UI - sophisticated 3D composition tools

Asset Library
Access to reusable components. Organize commonly used assets.

Timeline
Accurately define animation key frames similar to Flash or After Effects.

Inspector
Control fine details of presentation data. Create artist-friendly interactivity (Action/Events). Animate most any value.
UI Composer Architect

Information Architecture Platform
- Integrated Development Environment
- Modular component of UIC platform

Key Features
- State Machine IDE *Edit, Debug, Test*
- Lua Scripting IDE *Edit, Debug, Test*
- Device binding and management
UI Composer Viewer

OS Support
- Vibrante Linux & QNX
- Android
- Windows, Mac & Linux Desktop
- Portable codebase (GL & GLES)

Composition Engine
- Integrate with multiple visual sources
- Optimized 3D special effects and composition

Development Environment
- Lua scripting
- C/C++ Module support
Use Cases

- Automotive Instrument Cluster
- Automotive In-Vehicle Infotainment
- Project SHIELD
Automotive Instrument Cluster

Virtual “sky’s the limit” speedometer++

HMI Principles

- Sexy
- Glanceable
  - Simple/Familiar Material and Content
  - Consistent, Available Placement
  - Mode-Customized Appearance (e.g. Comfort vs. Sport)
  - Consistent Color Language (e.g. yellow is warning, red is bad)

Low Distraction

- Analog Gauges, Gentle Animation, Localized Animation
- Subtle Visual Cues (color gradients)
- Avoid “Fun” Content

Robust/Secure

- Accurate, Real-time Values
- ISO26262/ASIL-B
Automotive IVI

Center console: HVAC, Navigation, Music, Phone, Rear Camera, Apps?

HMI Principles
- Sexy
- Instantly Navigable
- Multimodal (buttons, jog dial, touch, voice, gestures)
- Identifiable Sensor Fusion (provably correct, integrated data display)
- Dynamic Data
Automotive IVI Demo
Project SHIELD

- **Portable Gaming Device**
  - Play games installed locally
  - Play games rendered remotely
  - Find and buy new games

- **HMI Principles**
  - Sexy
  - Seamless reentry
  - Multimodal (touch and controller)
  - Discoverable Navigation
  - Distinguish Functionality
  - Dynamic Data
Project SHIELD Demo
For More Info

- Want UI Composer? http://uicomposer.nvidia.com/
- Want to yell at me? gkistner@nvidia.com
- Questions?