presented by

American Megatrends

Creating Advanced Graphics Libraries on top of GOP

UEFI Summer Summit – July 16-20, 2012
Presented by Zachary Bobroff (AMI)

Agenda

- Introduction
- Implementing an Advanced Graphics Library
- Example Use Cases and Demonstration
- Call to Action

Introduction

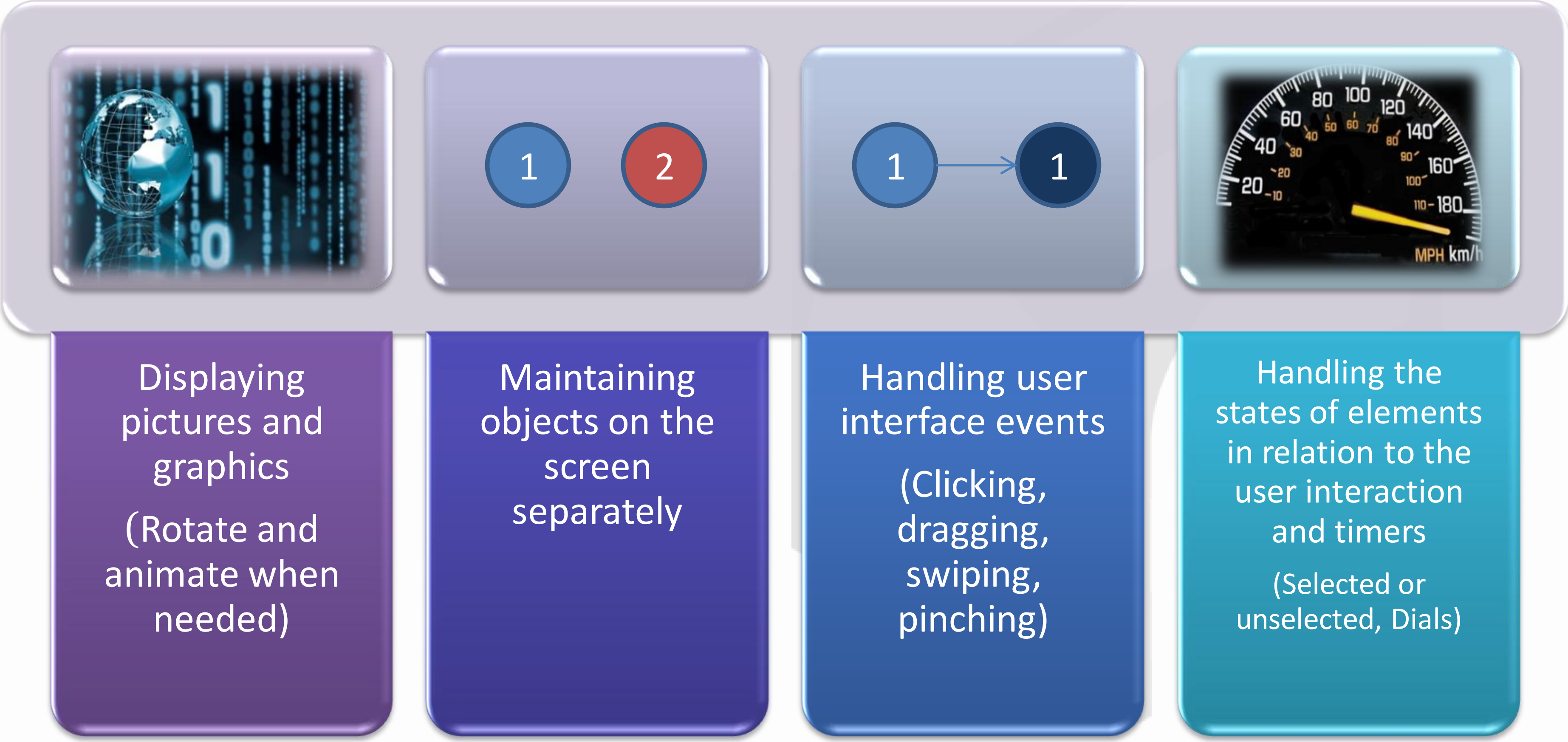
Introduction

Why is an advanced graphics library useful?

- An advanced graphics library has advantages:
  - Computers are more than just traditional PCs these days and may not have traditional input devices like keyboards and mice
  - The use of smart phones and tablets have changed the way people think about user interfaces
  - Using an advanced graphics library will allow you to create a unique and feature rich user interface
  - The user interface can be created to be intuitive and can even contain information that would traditionally be in a user manual

Smart Art Example

An advanced graphics library capabilities include:



Displaying pictures and graphics

(Rotate and animate when needed)

Maintaining objects on the screen separately

Handling user interface events

(Clicking, dragging, swiping, pinching)

Handling the states of elements in relation to the user interaction and timers

(Selected or unselected, Dials)

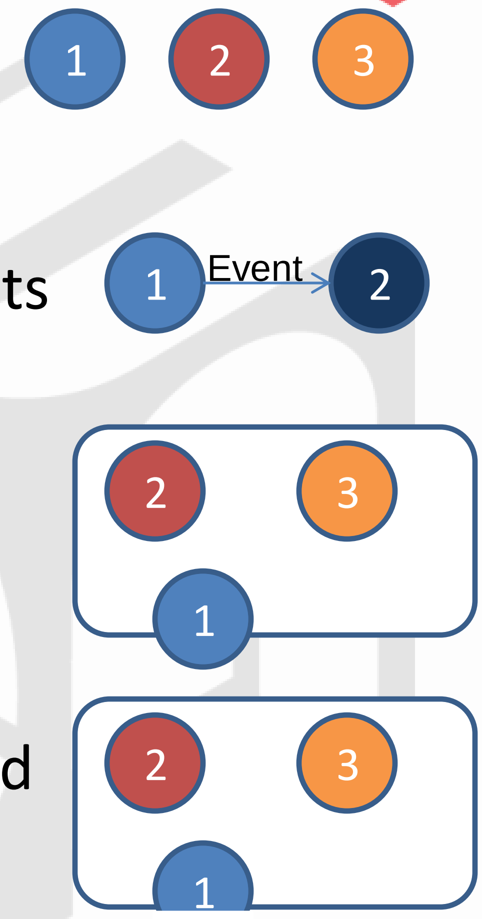
What UEFI Provides

- UEFI provides the basic building blocks to create everything an advanced graphics library needs
- Familiar elements to most UEFI developers will be:
  - Simple text in
  - Simple text out
  - Simple pointer protocol
  - Absolute pointer protocol
  - Graphics output protocol
  - Event infrastructure for timers and user input

Implementing an Advanced Graphics Library

Graphics Library Building Blocks

- Viewable constructs – a common item which abstracts the events and display functionality
- Event handling – engine which prioritizes events and updates the viewable constructs for each event type (user input, timer functions, etc)
- Viewable window – the area of viewable constructs that will be rendered on the screen
- Rendering engine – engine that will generate the final displayable screen based on the current set of viewable constructs



Connections with the UEFI Specification

- View constructs – can be protocols all with the same GUID and common functions
- Event handling engine – use EFI timer event, and protocols for simple text in, absolute pointer and simple pointer
- Rendering engine – takes the entire displayable buffer and does a BLT for the entire GOP buffer all at one time
  - Doing an entire screen buffer at once limits artifacts being seen

States and Animations

- All viewable constructs contain a display function that returns a BLT
- All viewable constructs can have an event handler function that is notified on user input and timer events
  - The display function will check internal states of the viewable construct, which can be updated by the handler function, and will change how the BLT is constructed
- The combination of the event handler functions and the display functions allows the viewable construct to gradually change the display or animate itself

Event Engine

- The event engine is a function that will be notified on user input or timer expiration
  - The event engine then calls the appropriate handler function
- On user input, it will call a handler for the currently selected construct
- On timer expiration, all viewable constructs will be notified

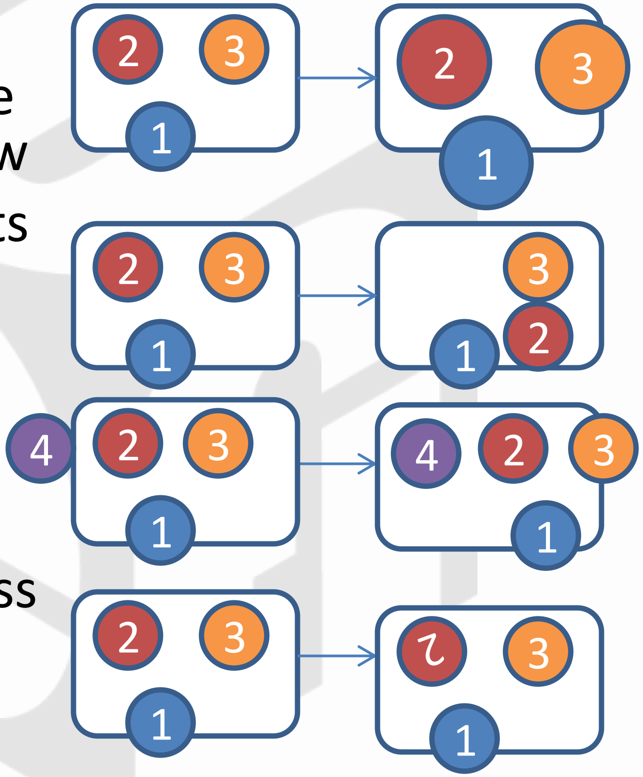
Rendering Engine

- The rendering engine is a function called at whatever frequency the application desires the screen to be redrawn
- The rendering engine takes all viewable constructs within the current viewable window and calls their current BLT function
- The individual BLTs of the viewable constructs are then compiled into a screen wide BLT and this entire BLT is written to the GOP protocol

Doing an entire screen BLT implements basic double buffering even if the GOP does not support it

Extra Advanced Graphics Features

- Due to smart phones and tablets, users have become accustomed to certain user interface features including:
  - Zoom capabilities– Just scale the viewable constructs in the current viewable window
  - Drag and drop – Move viewable constructs location within the current viewable window
  - Swipe gestures – Move the viewable window's location and show different viewable constructs
  - Object rotation – As the user swipes across a viewable construct, its BLT function returns views from different angles



Example Use Cases and Demonstration

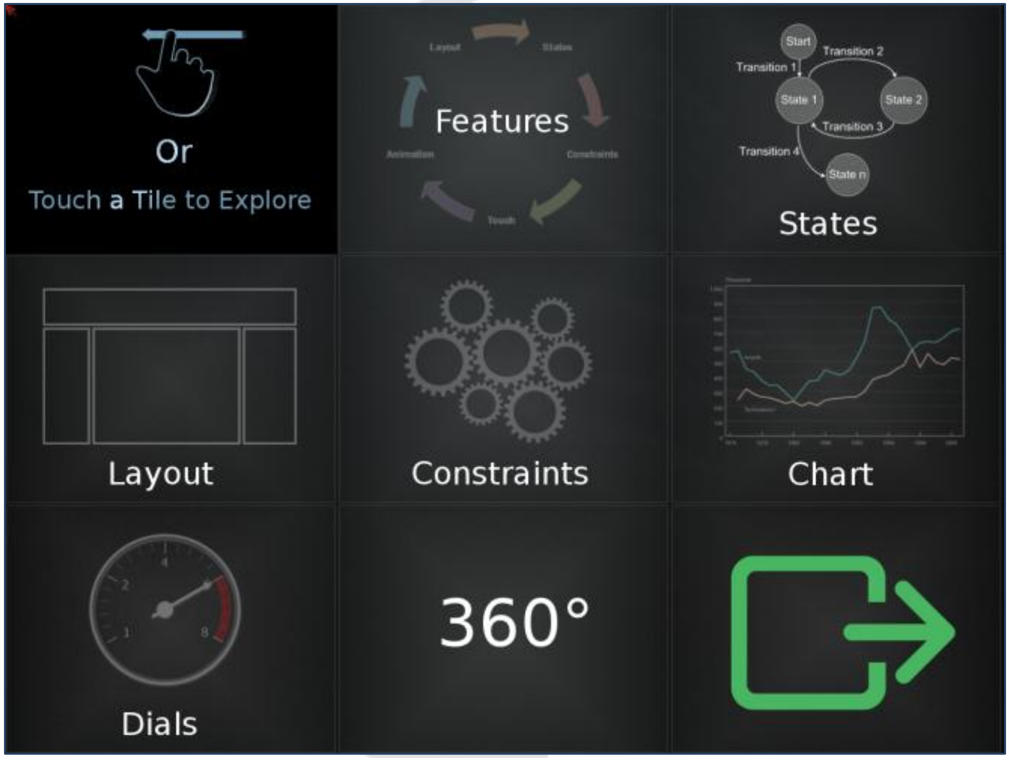
Use Cases for Advanced Graphics

- A graphics library allows an application to:
  - Have PowerPoint like effects
  - Transition between object states
  - Have easily switchable layouts
  - Use constraints for animation
  - Display charts
  - Have dials
  - Use 360 degree view of objects

Demonstration

- Demonstration of sample effects

American
Megatrends



Further Expansion and Demo

- With the building blocks discussed previously, a full graphics library like OpenGL can be ported to EFI
- Going to the level of OpenGL support allows complex 3D applications written for an OS to be moved into EFI

Call to Action

Call to Action

- Users are expecting a more interactive experience than traditional BIOS setup provides
- Creating intuitive user interfaces can decrease the amount of support questions received from customers
- Putting the entire manual in the pre-boot environment decreases production costs

Thanks for attending the
UEFI Summer Summit 2012

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

American
Megatrends