Developing Visual Interfaces for Mobile Devices

Ben Watson
NC State U

Vidya Setlur
Nokia Research

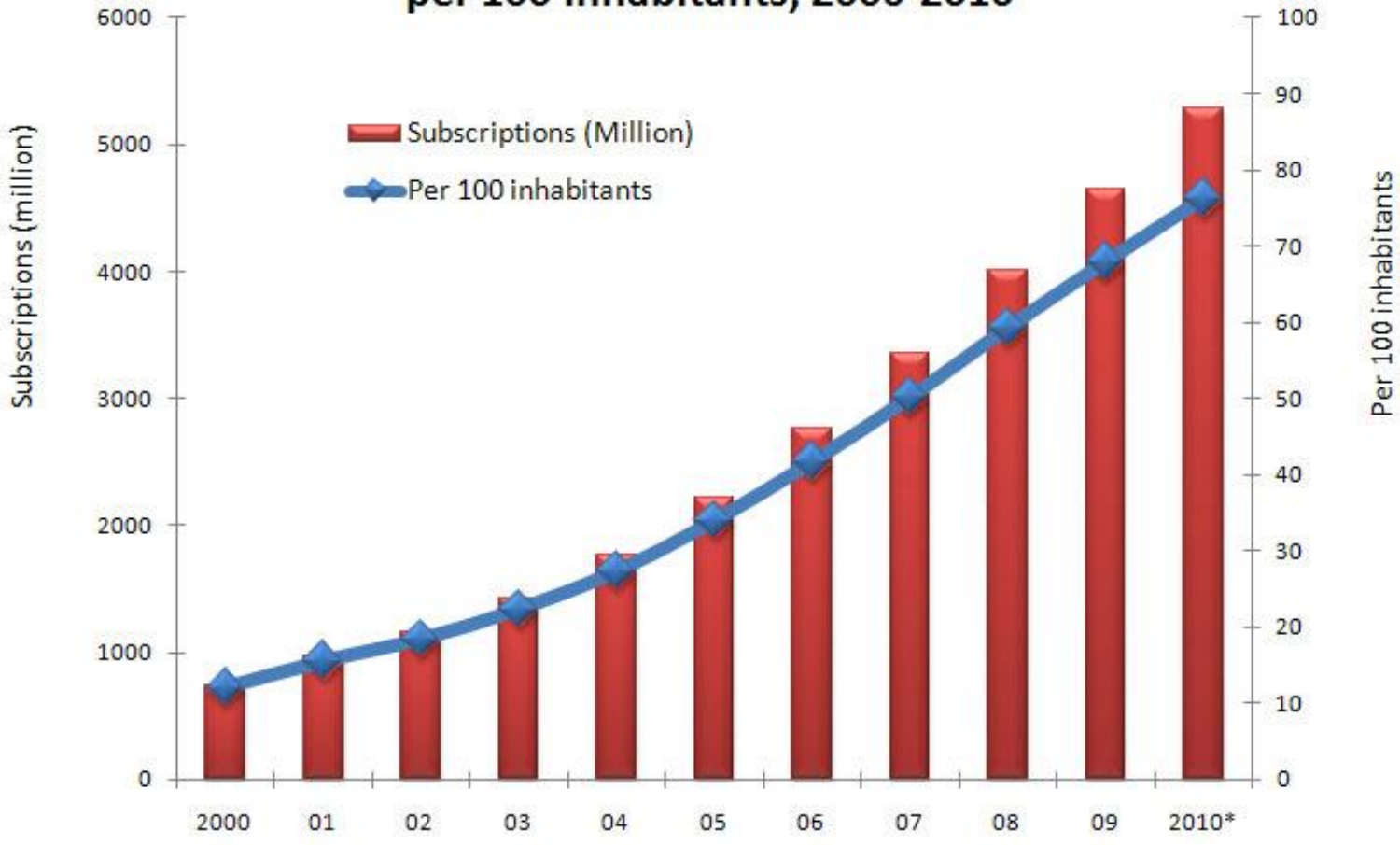
Kari Pulli
NVIDIA

introduction
UIs
graphics

introduction

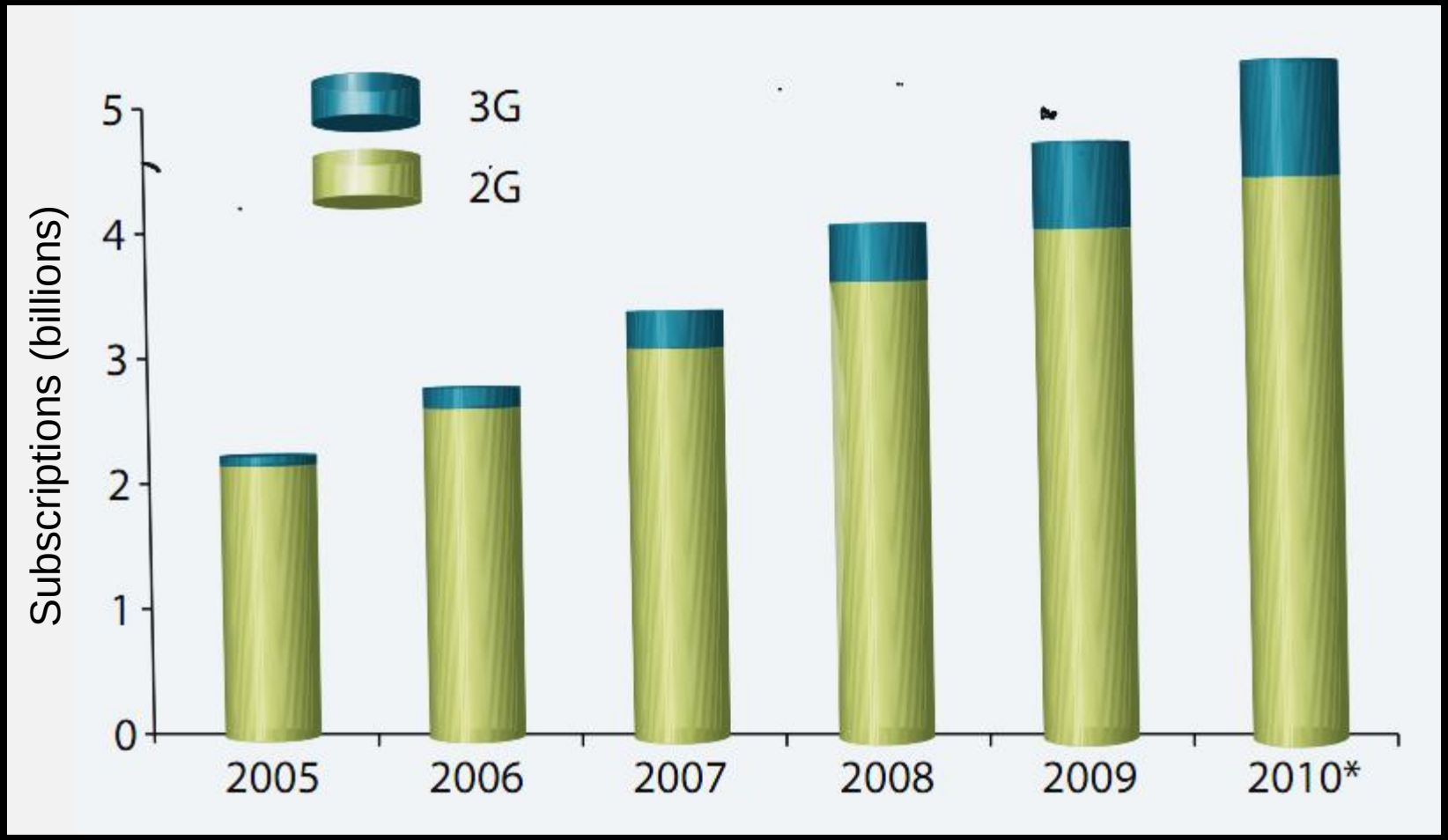
why mobiles?

Global mobile cellular subscriptions, total and per 100 inhabitants, 2000-2010



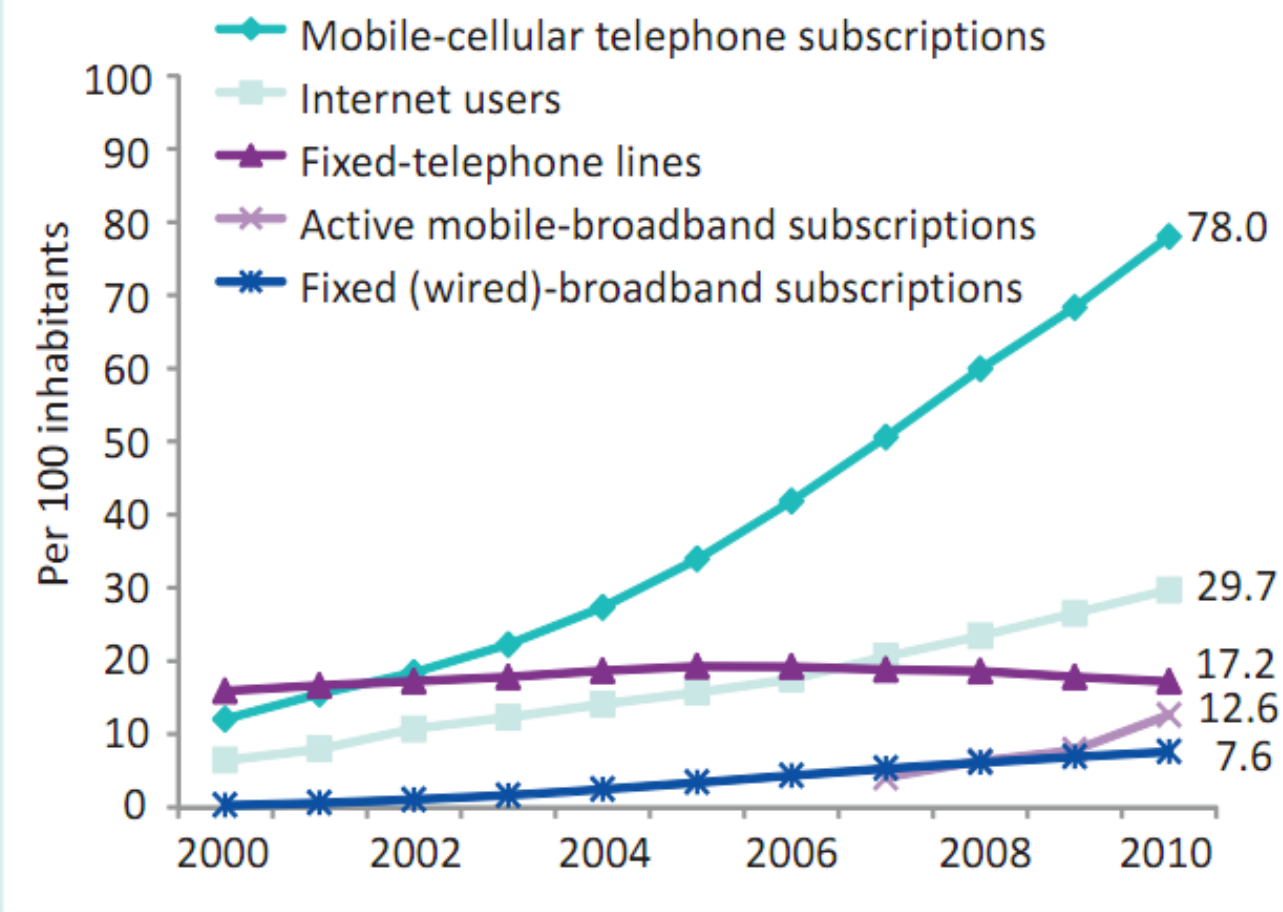
*Estimates

Source: ITU World Telecommunication /ICT Indicators database


Subscriptions (billions)
5
4
3
2
1
0
3G
2G
2005
2006
2007
2008
2009
2010*

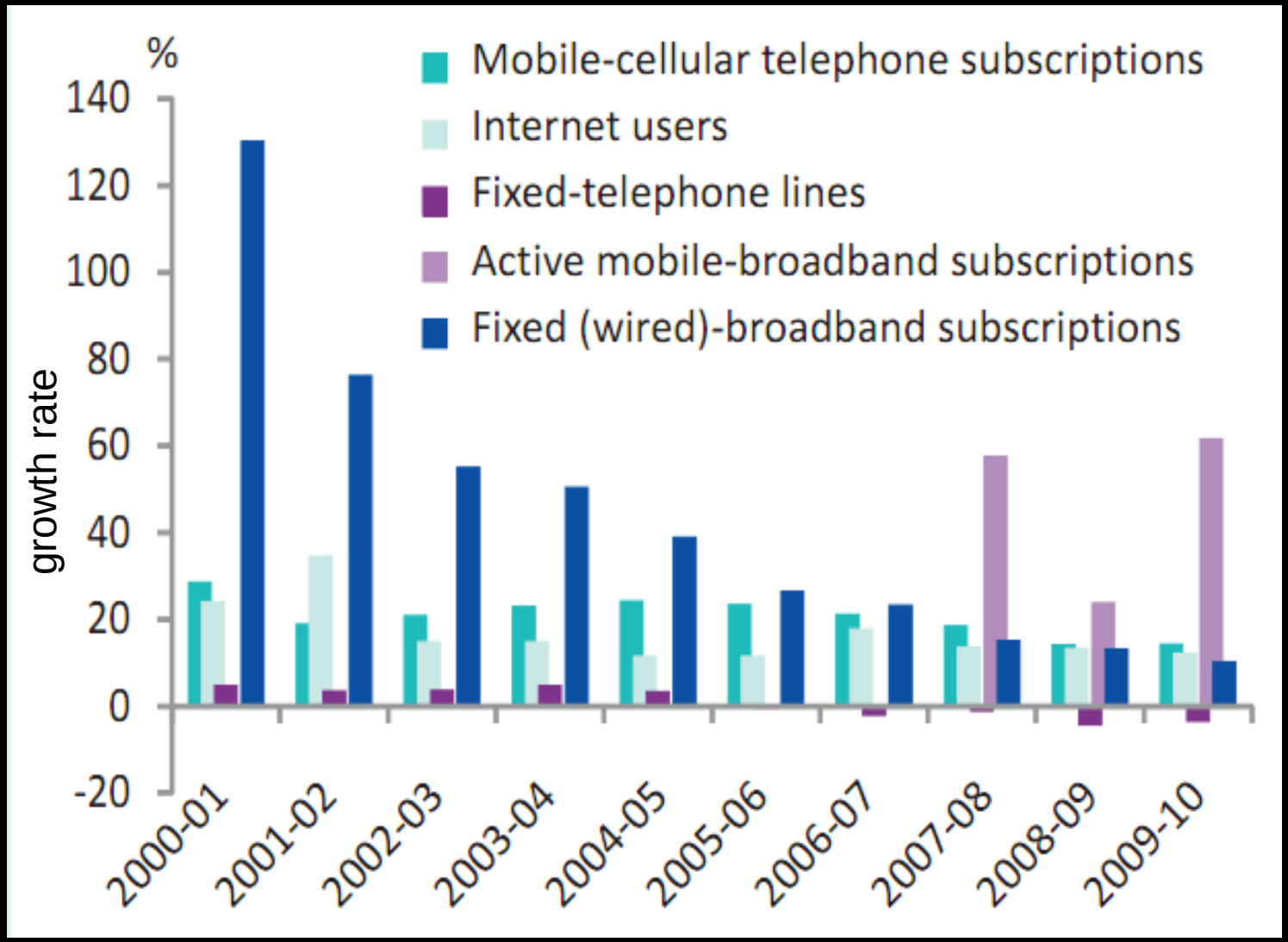
source: ITU

vs. technology



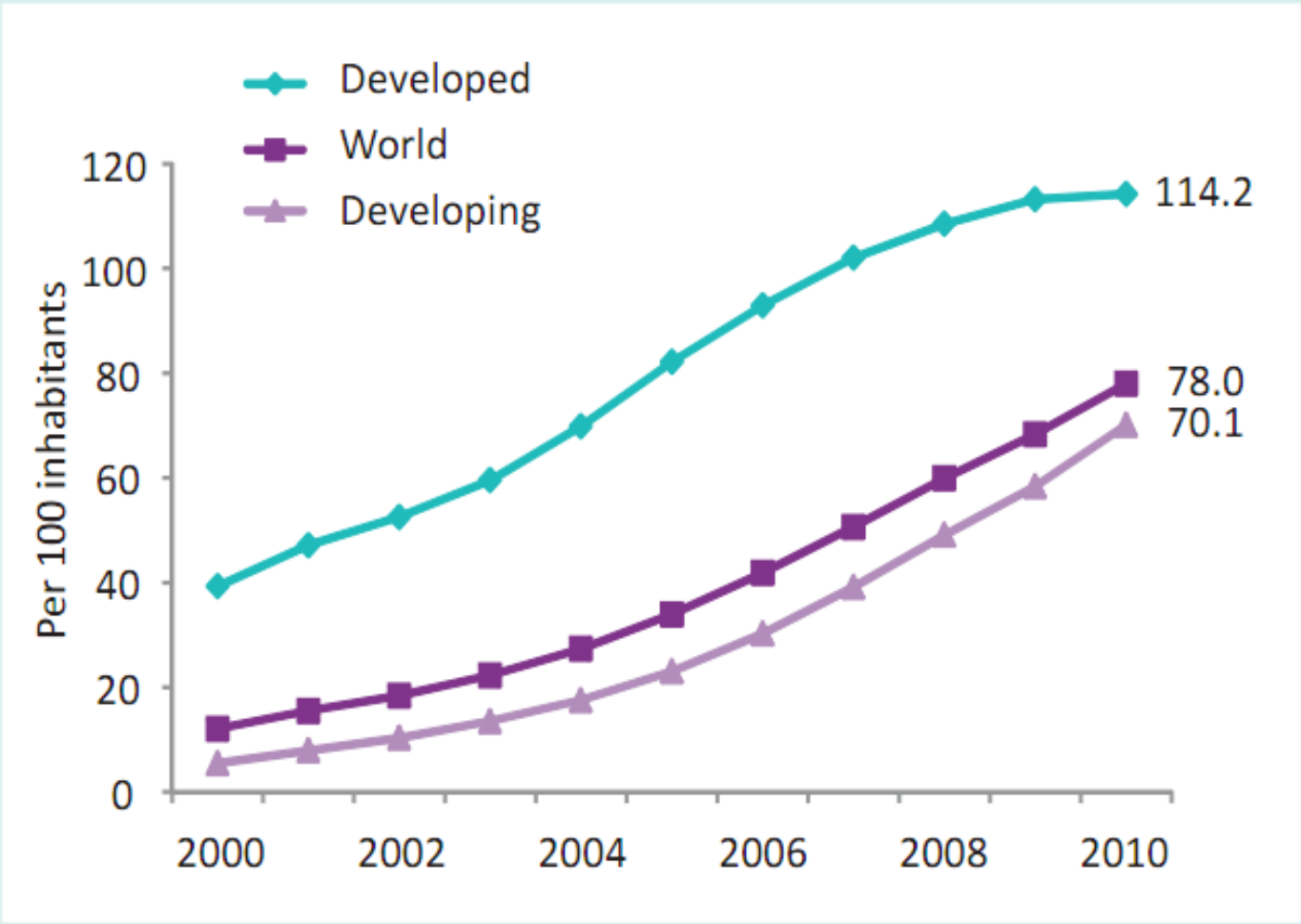
Source: ITU World Telecommunication/ICT Indicators database.

vs. technology



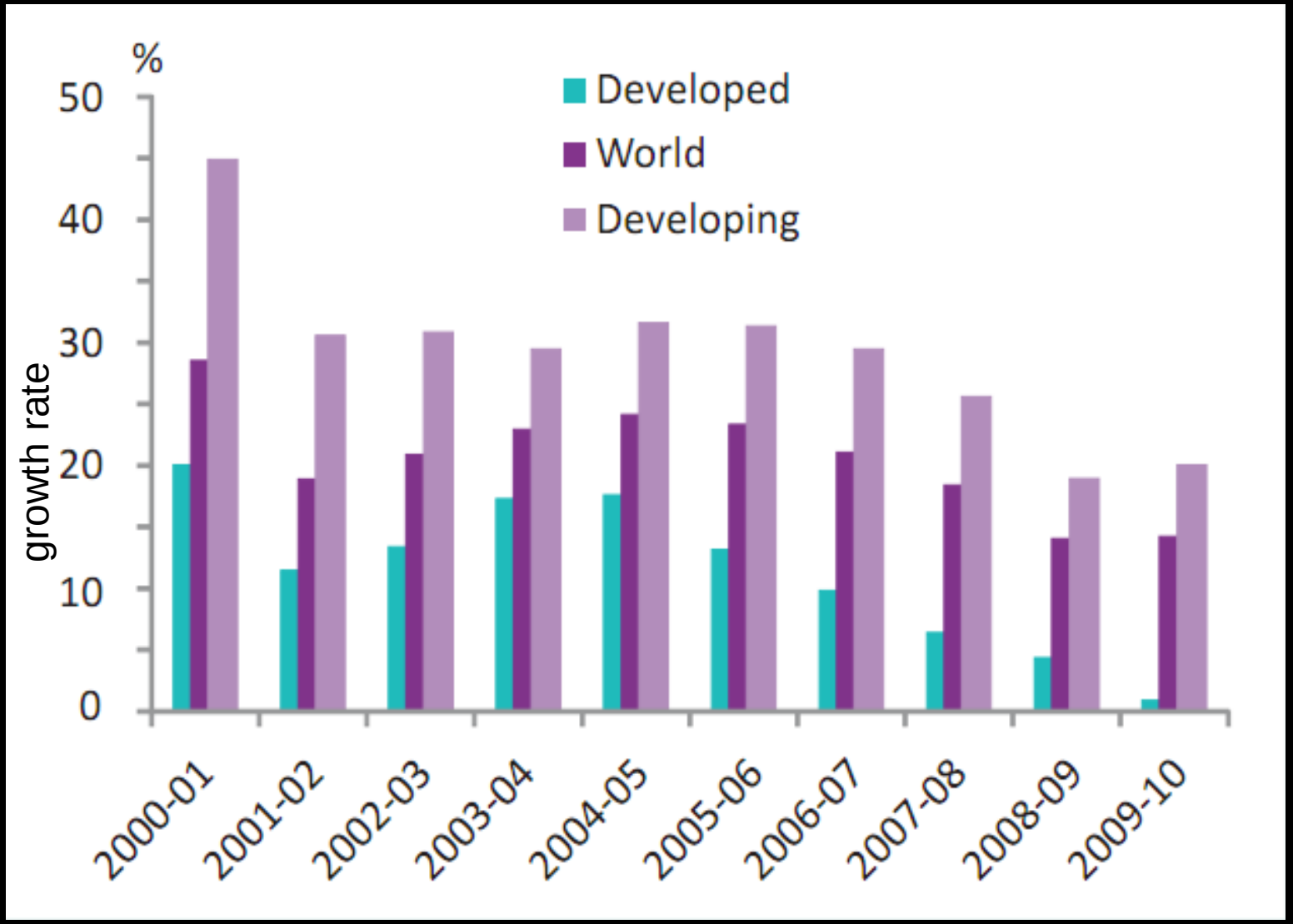
source: ITU

vs. development



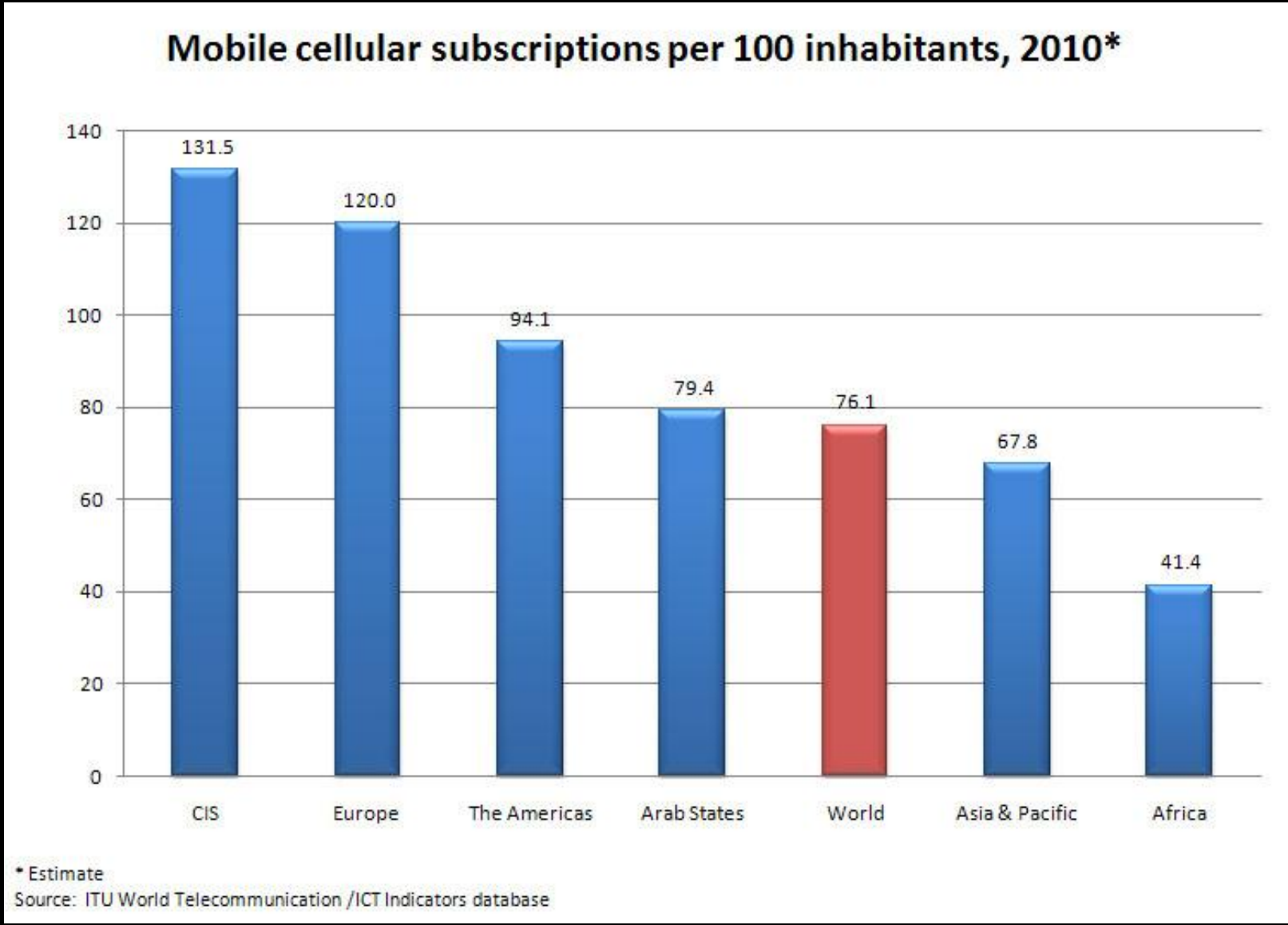
Source: ITU World Telecommunication/ICT Indicators database.

vs. development

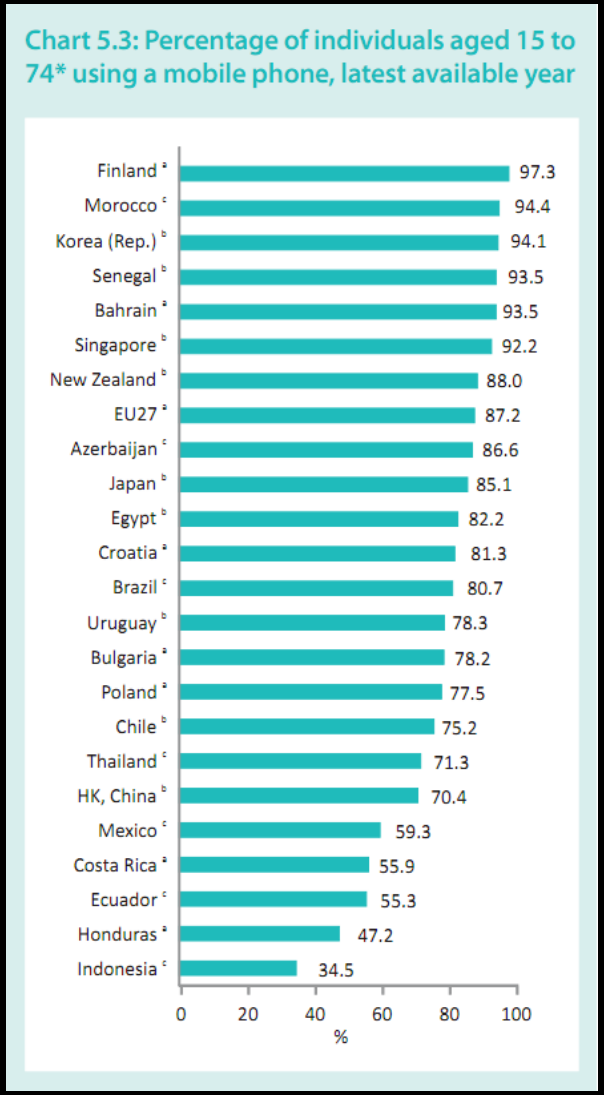


source: ITU

vs. region

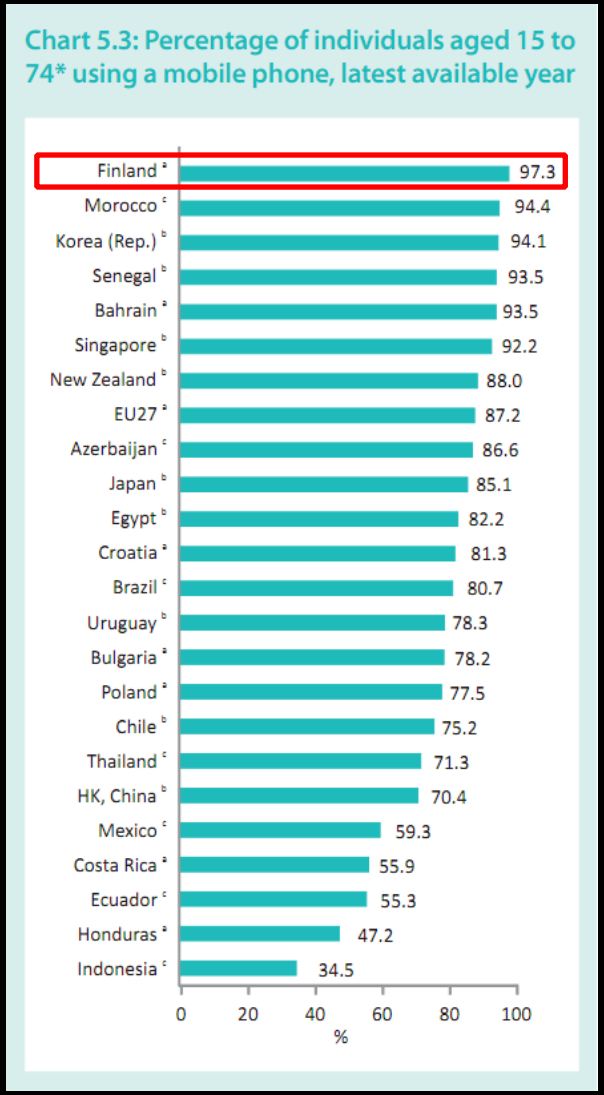

Mobile cellular subscriptions per 100 inhabitants, 2010*
140
120
100
80
60
40
20
0
131.5
120.0
94.1
79.4
76.1
67.8
41.4
CIS
Europe
The Americas
Arab States
World
Asia & Pacific
Africa
* Estimate
Source: ITU World Telecommunication /ICT Indicators database

vs. nation



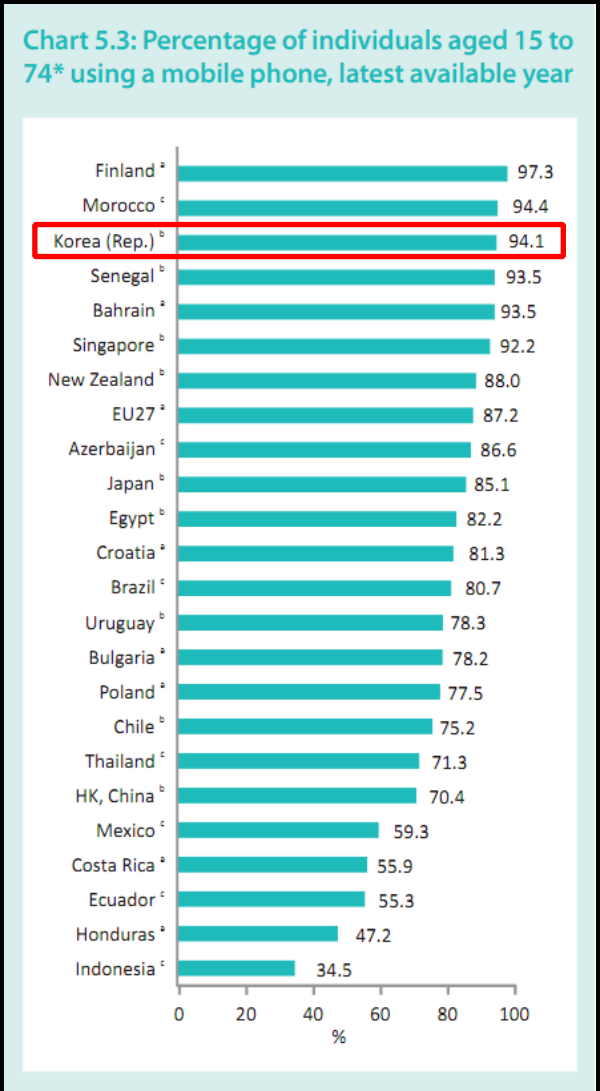
source: ITU

vs. nation



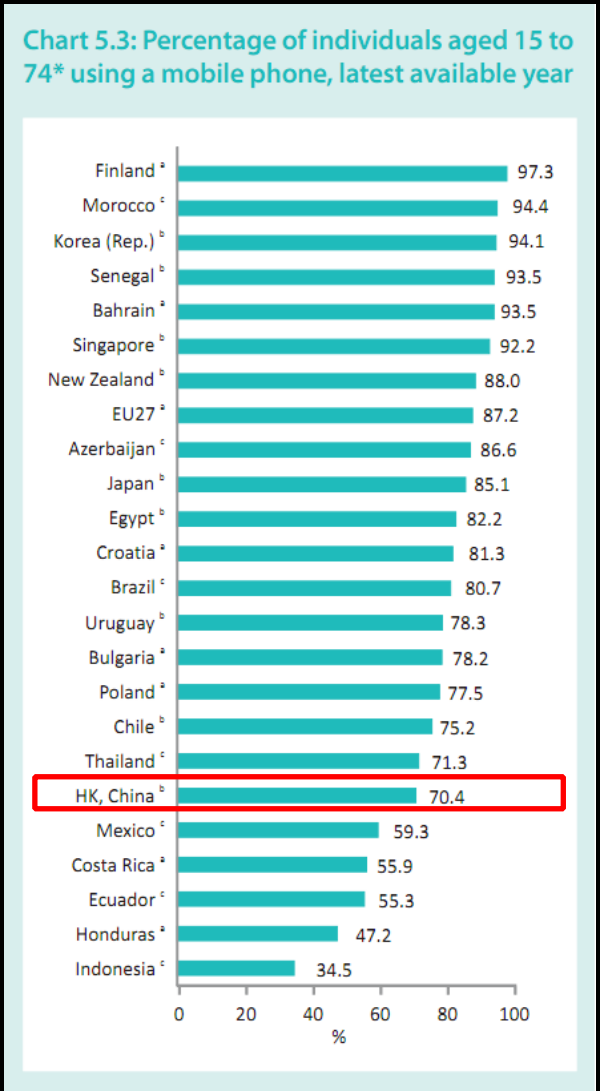
source: ITU

vs. nation



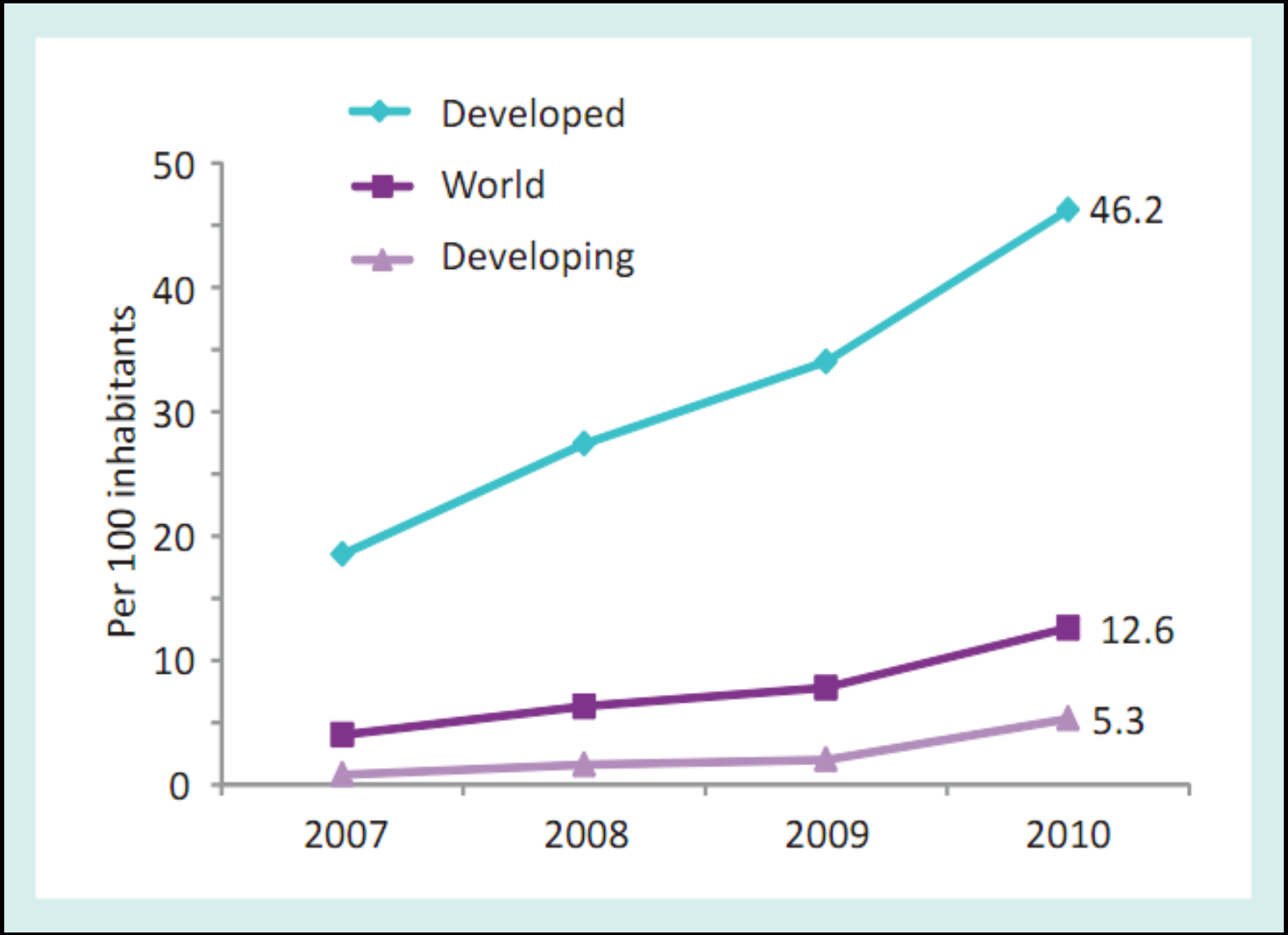
source: ITU

vs. nation



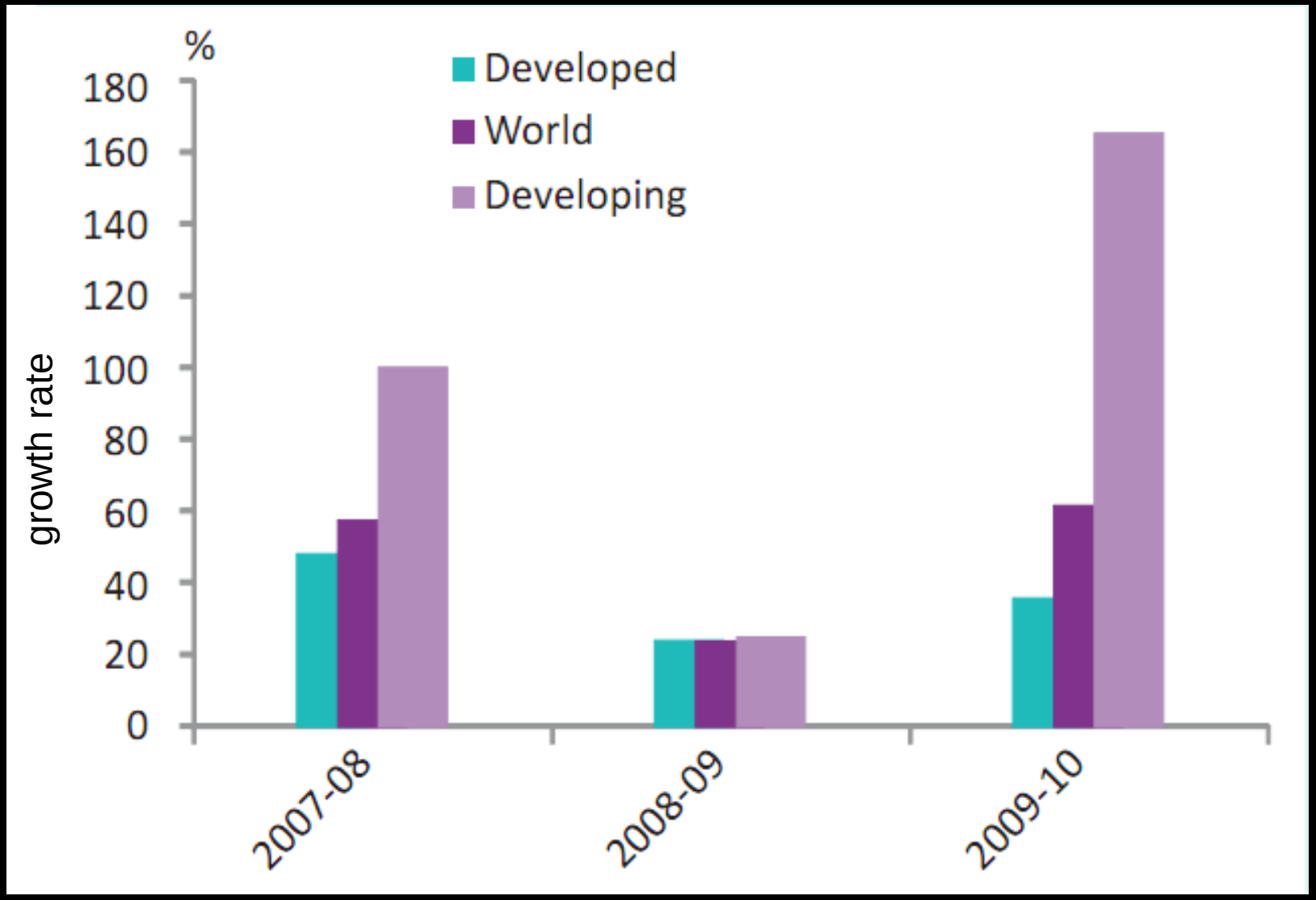
source: ITU

mobile broadband vs. development



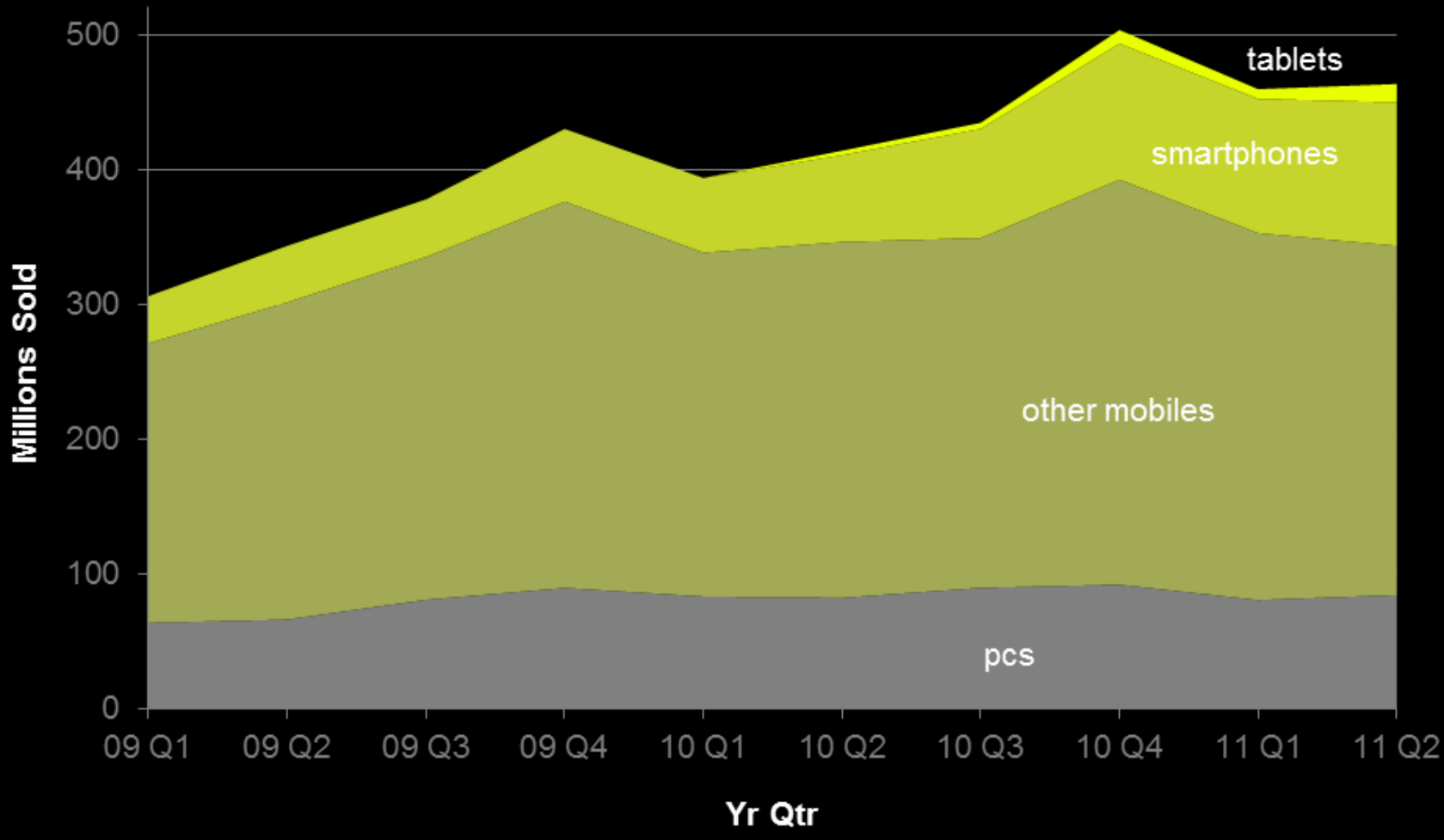
source: ITU

mobile broadband vs. development



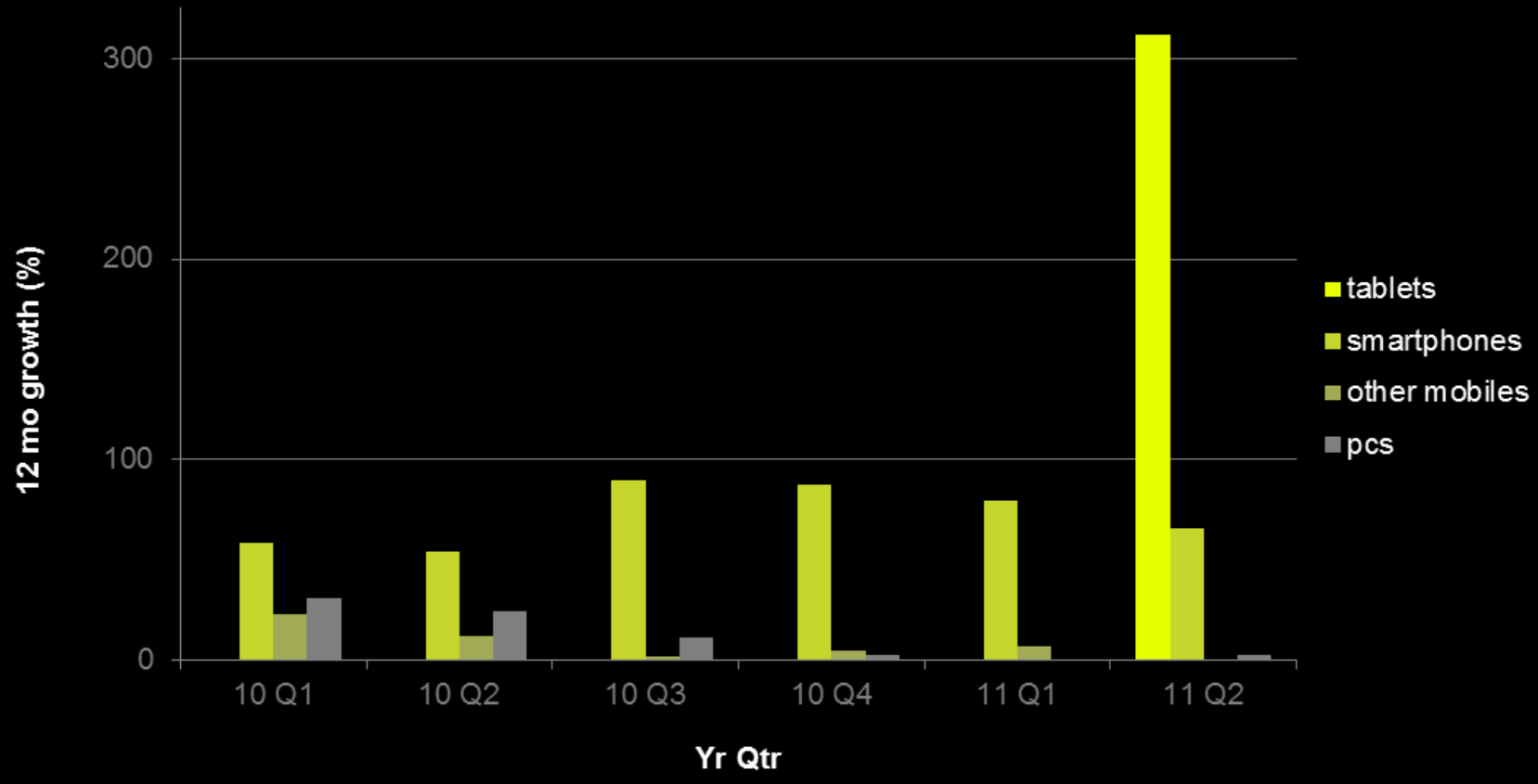
source: ITU

vs. PCs



data: IDC

vs. PCs



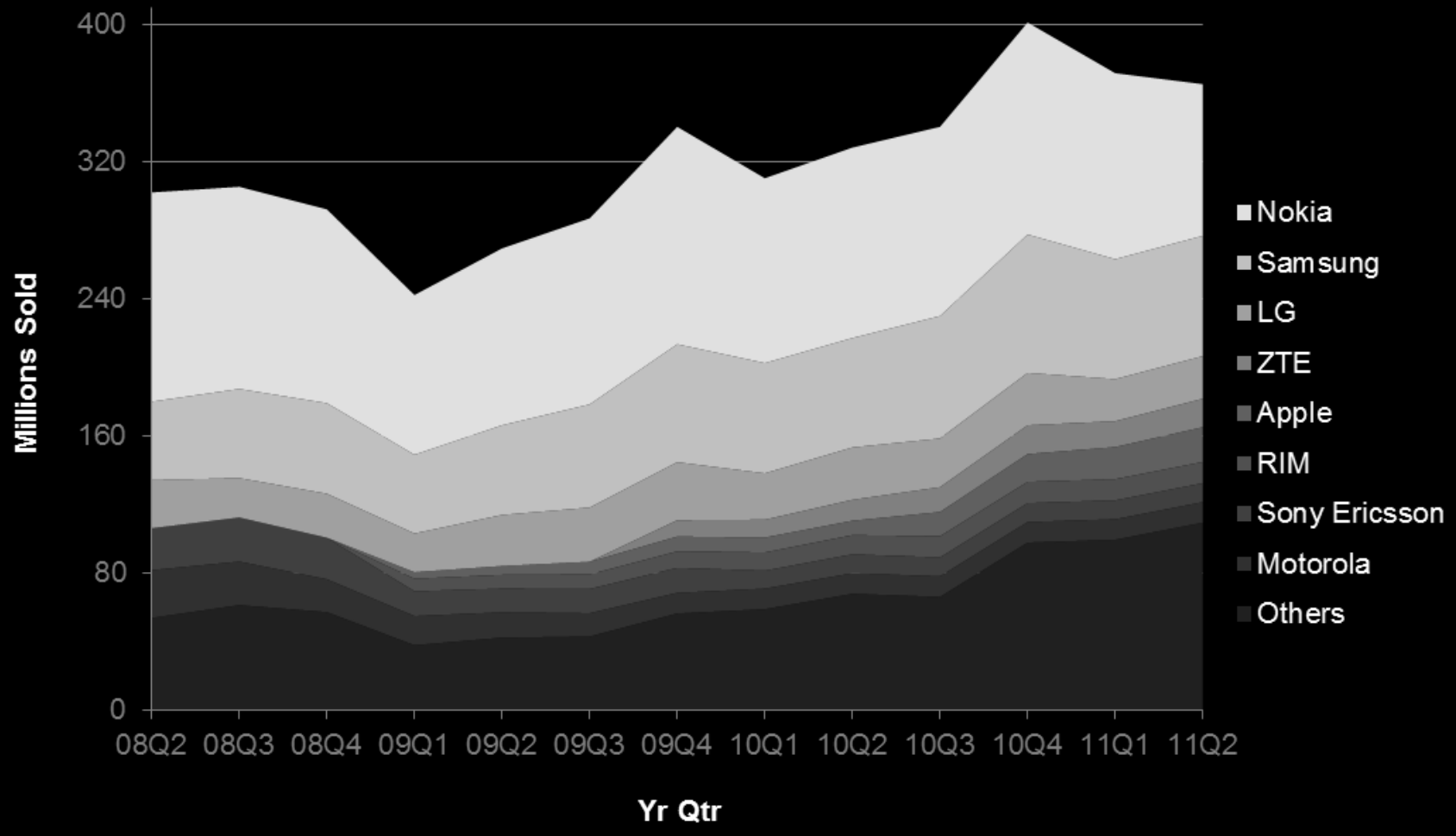
data: IDC


N
NE
E
SE
S
SW
W
NW

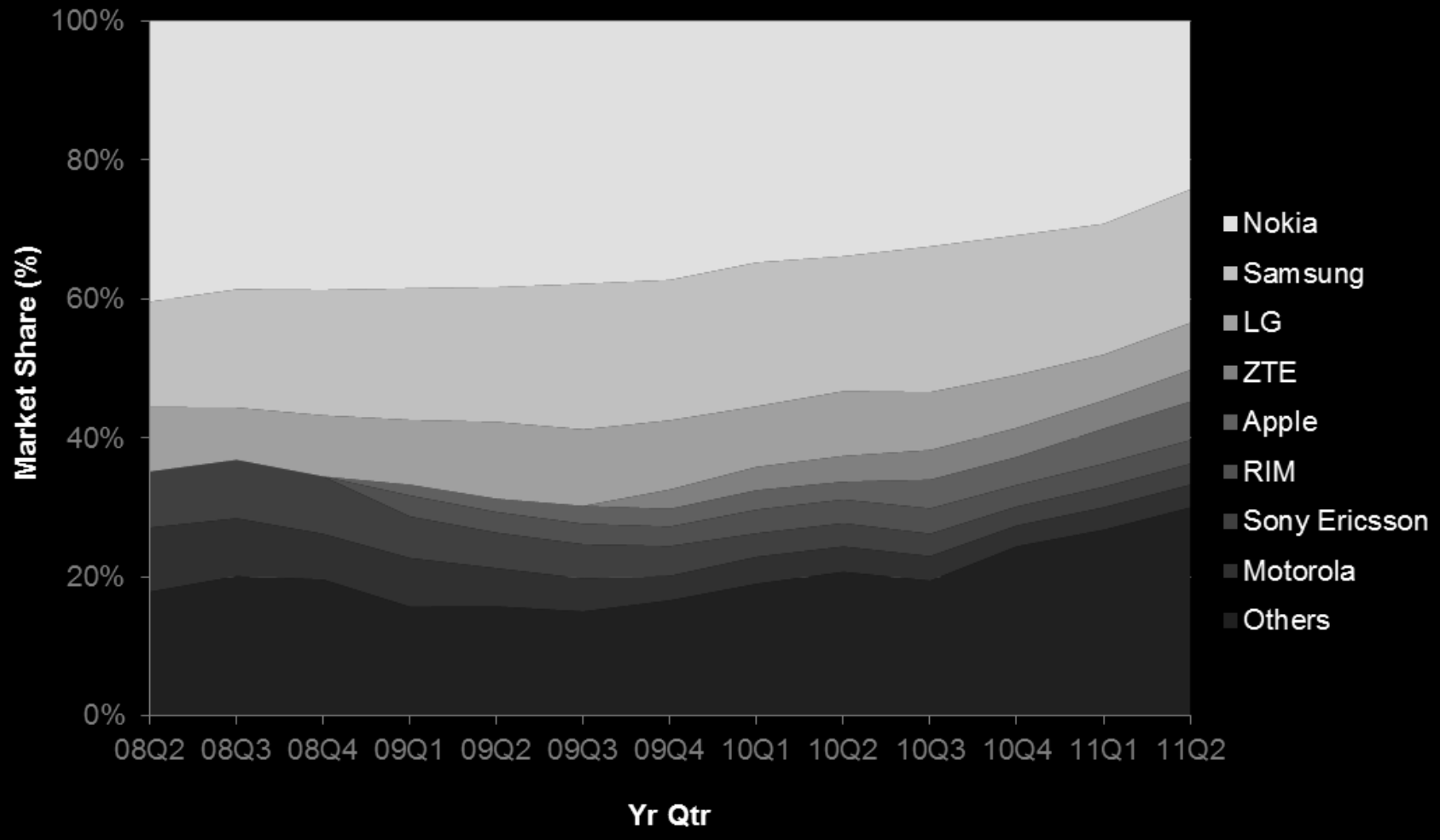

AT&T
9:42 AM
Cancel
Re: Jenny's Birthday
Send
To: Rob Tucker
Cc:
Subject: Re: Jenny's Birthday
Great pi
pick
Q W E R T Y U I
A S D F G H J
Z X C V B N M
.?123
space
return

VS

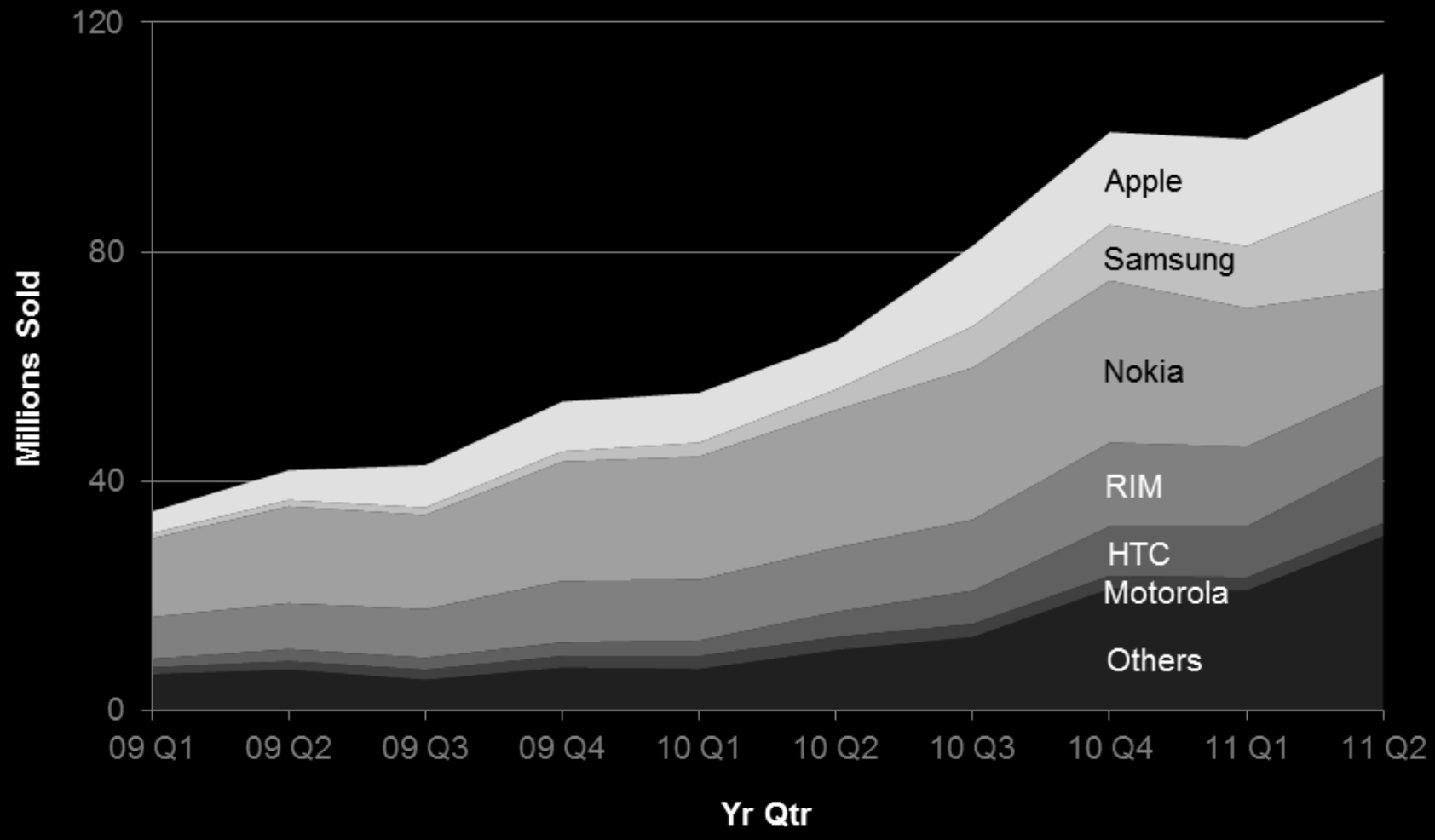
the mobile field


400
320
240
160
80
0
Millions Sold
08Q2
08Q3
08Q4
09Q1
09Q2
09Q3
09Q4
10Q1
10Q2
10Q3
10Q4
11Q1
11Q2
Yr Qtr
Nokia
Samsung
LG
ZTE
Apple
RIM
Sony Ericsson
Motorola
Others

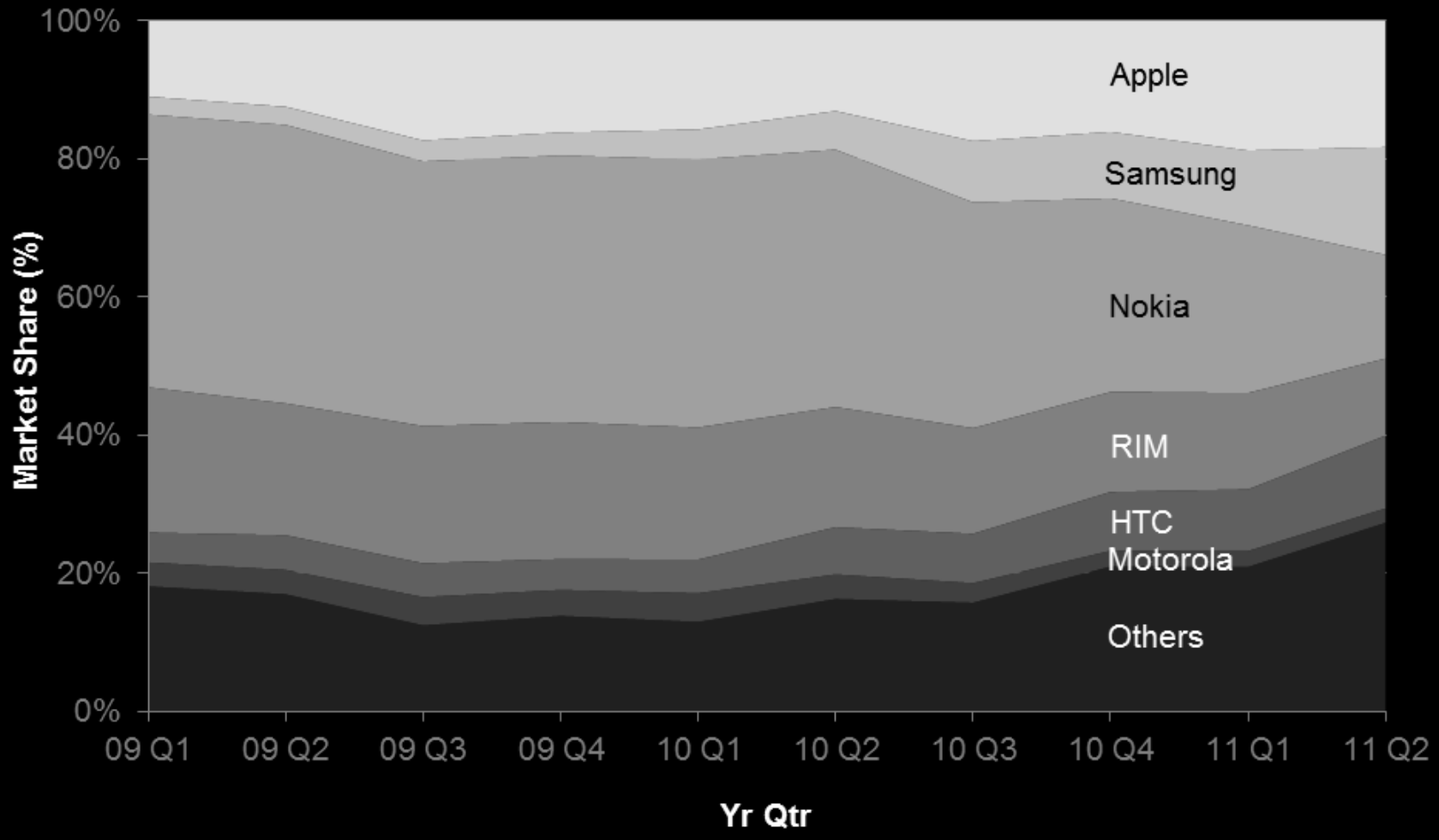
data: IDC


100%
80%
60%
40%
20%
0%
Market Share (%)
08Q2
08Q3
08Q4
09Q1
09Q2
09Q3
09Q4
10Q1
10Q2
10Q3
10Q4
11Q1
11Q2
Yr Qtr
Nokia
Samsung
LG
ZTE
Apple
RIM
Sony Ericsson
Motorola
Others

data: IDC


120
80
40
0
Millions Sold
Apple
Samsung
Nokia
RIM
HTC
Motorola
Others
09 Q1
09 Q2
09 Q3
09 Q4
10 Q1
10 Q2
10 Q3
10 Q4
11 Q1
11 Q2
Yr Qtr

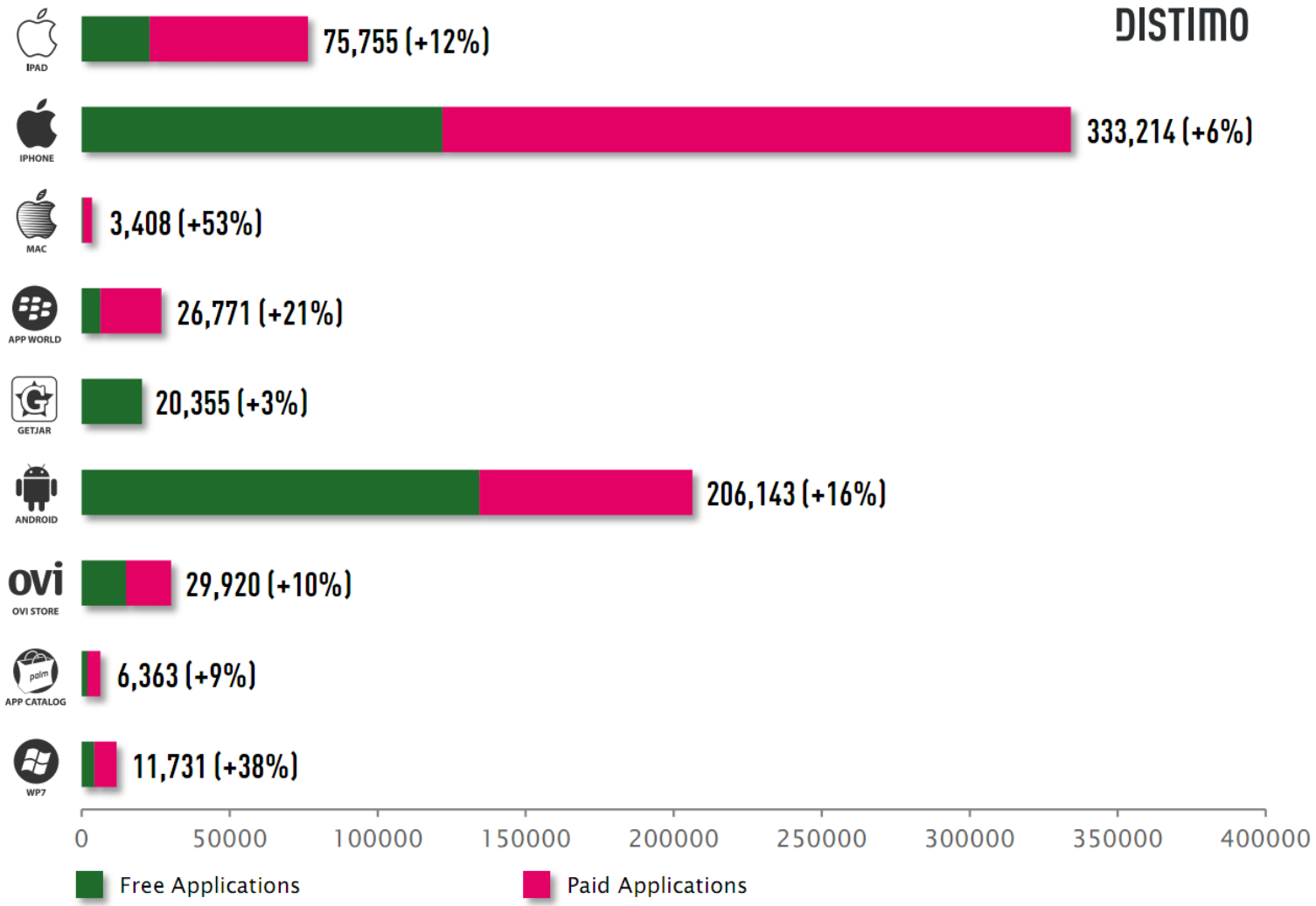
data: IDC


100%
80%
60%
40%
20%
0%
Market Share (%)
Apple
Samsung
Nokia
RIM
HTC
Motorola
Others
09 Q1
09 Q2
09 Q3
09 Q4
10 Q1
10 Q2
10 Q3
10 Q4
11 Q1
11 Q2
Yr Qtr

data: IDC

NUMBER OF AVAILABLE APPLICATIONS

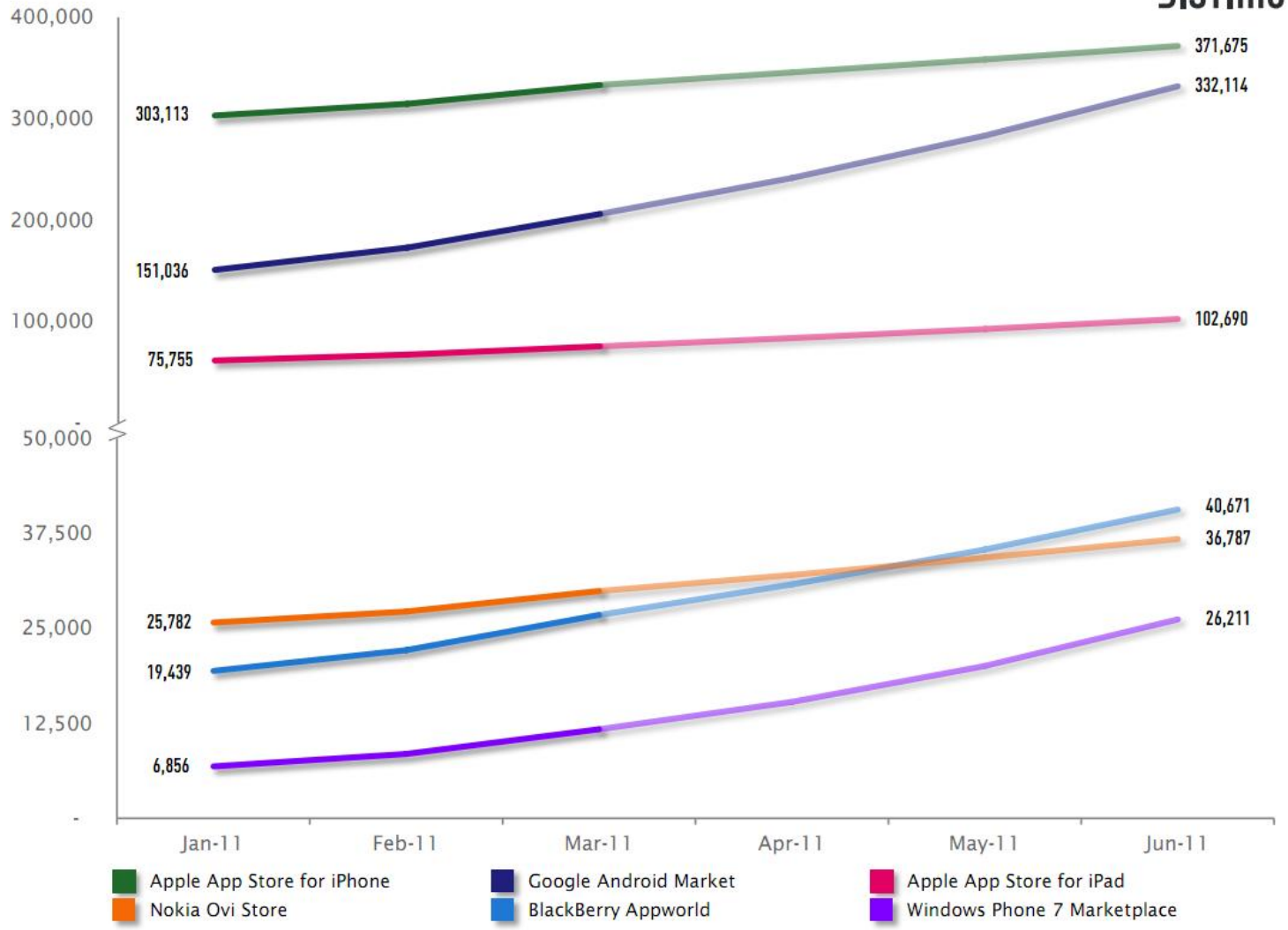
MARCH 2011 - UNITED STATES



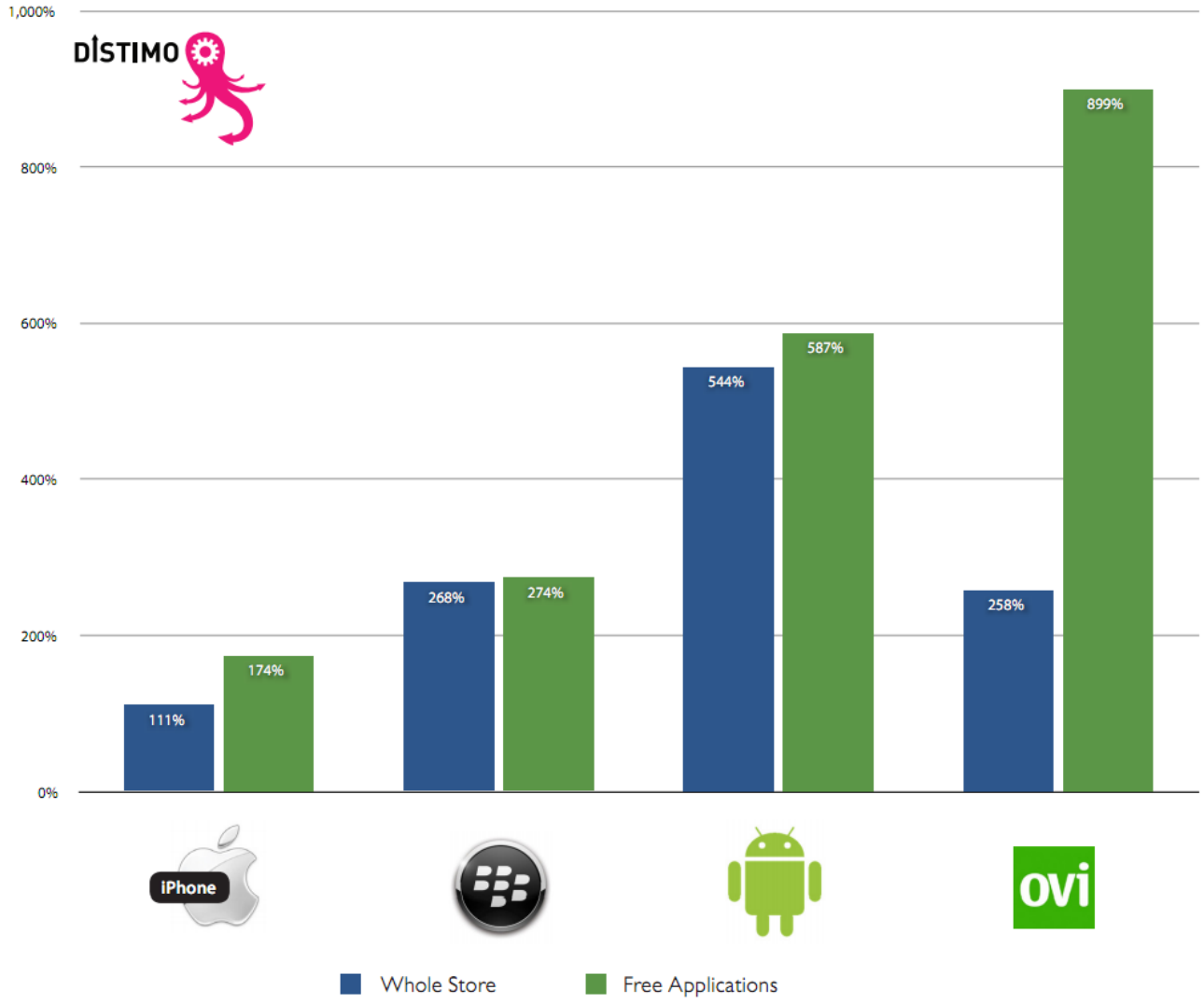
FUTURE DEVELOPMENTS IN THE APP STORE ECOSYSTEM

FORECAST FIRST HALF 2011 - UNITED STATES

DISTIMO

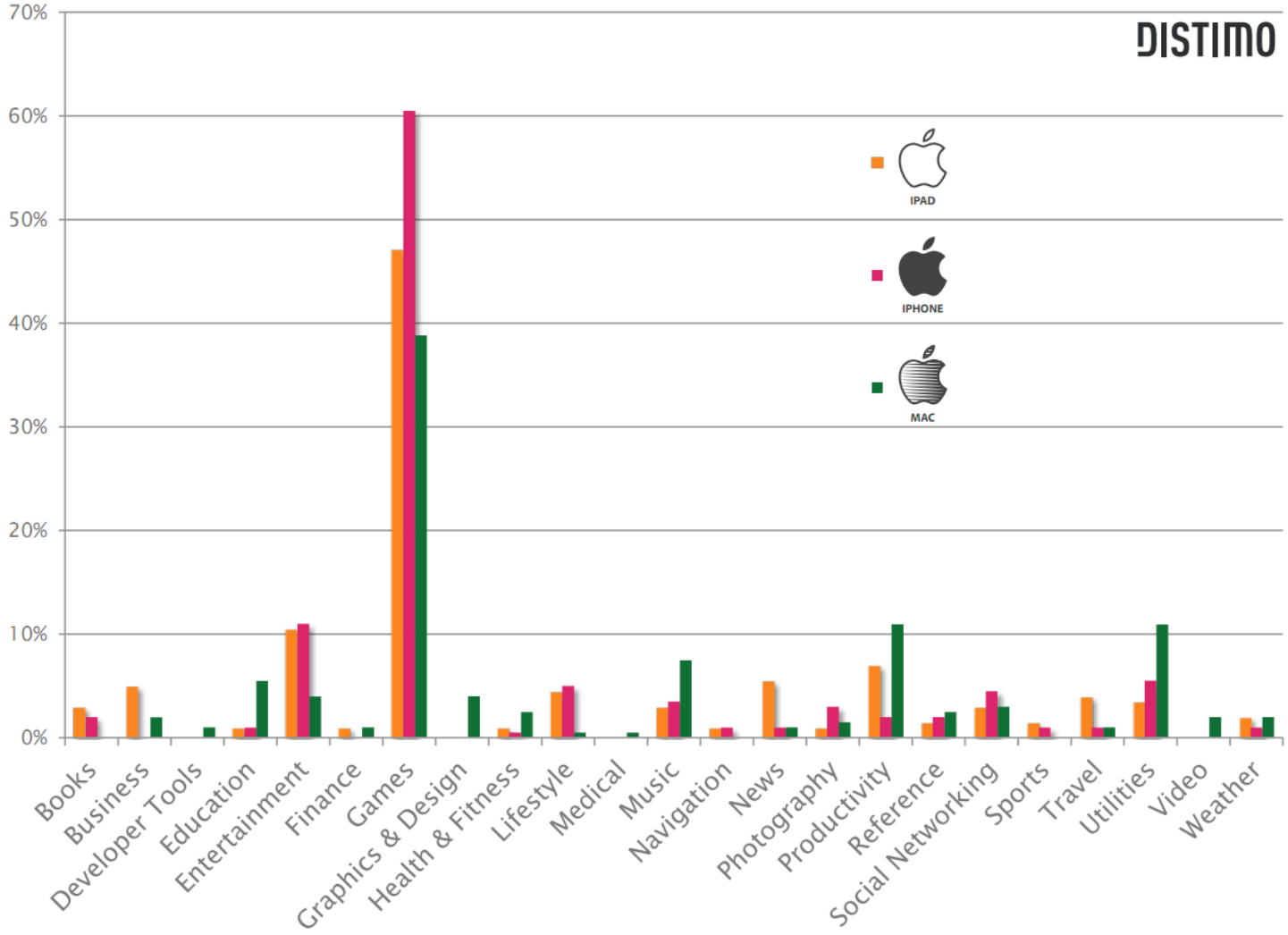


Application Store Growth (January - December 2010, United States)



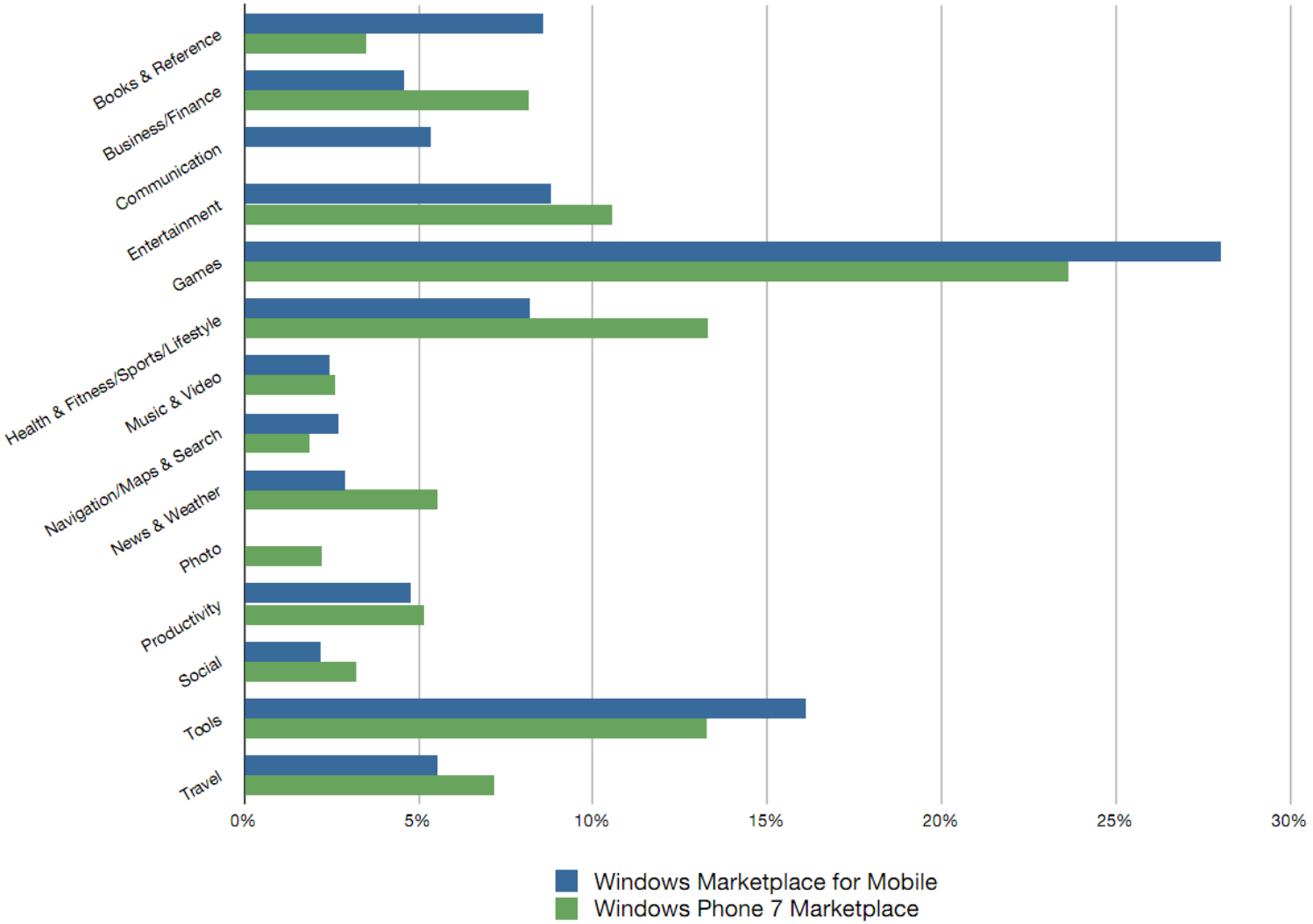
CATEGORY BREAKDOWN OF THE 100 MOST POPULAR APPLICATIONS

FEBRUARY 2011 - UNITED STATES



DISTIMO

Proportion Of Applications By Category (United States)



Top 5 growth Categories per Store

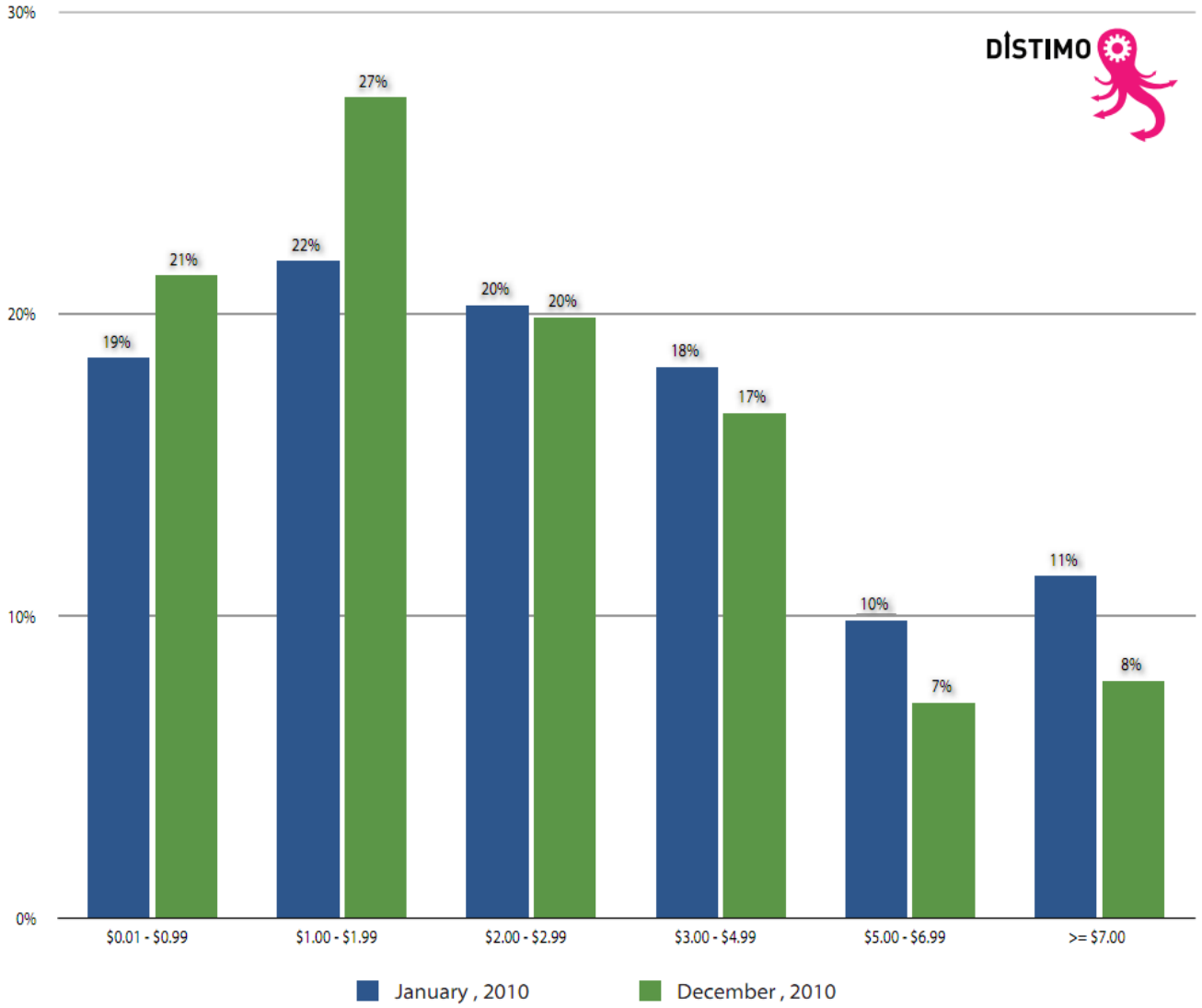
United States

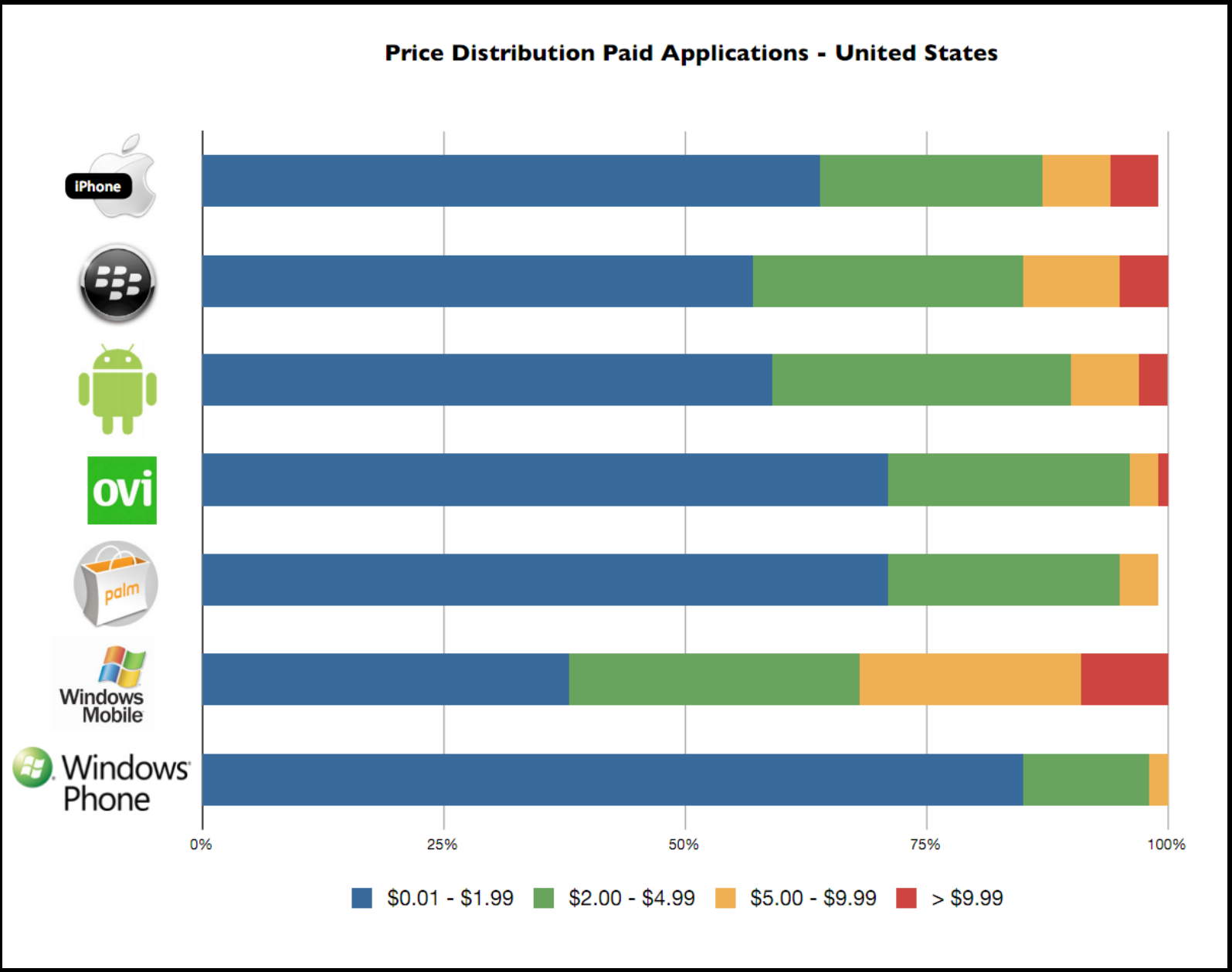
Apple App Store for iPhone	111% (average)	BlackBerry App World	268% (average)
Business	186%	Reference & eBooks	733%
Medical	156%	Themes	653%
Lifestyle	145%	Music & Audio	402%
Finance	144%	News	353%
Music	143%	Health & Wellness	243%

Google Android Market	544% (average)	Nokia Ovi Store	258% (average)
Application - Comics	802%	News & Info	1281%
Application - Sports	748%	Sports	1073%
Games - Card & Casino	664%	Business	909%
Application - Entertainment	589%	Music	861%
Application - Health	488%	City guides & Maps	743%

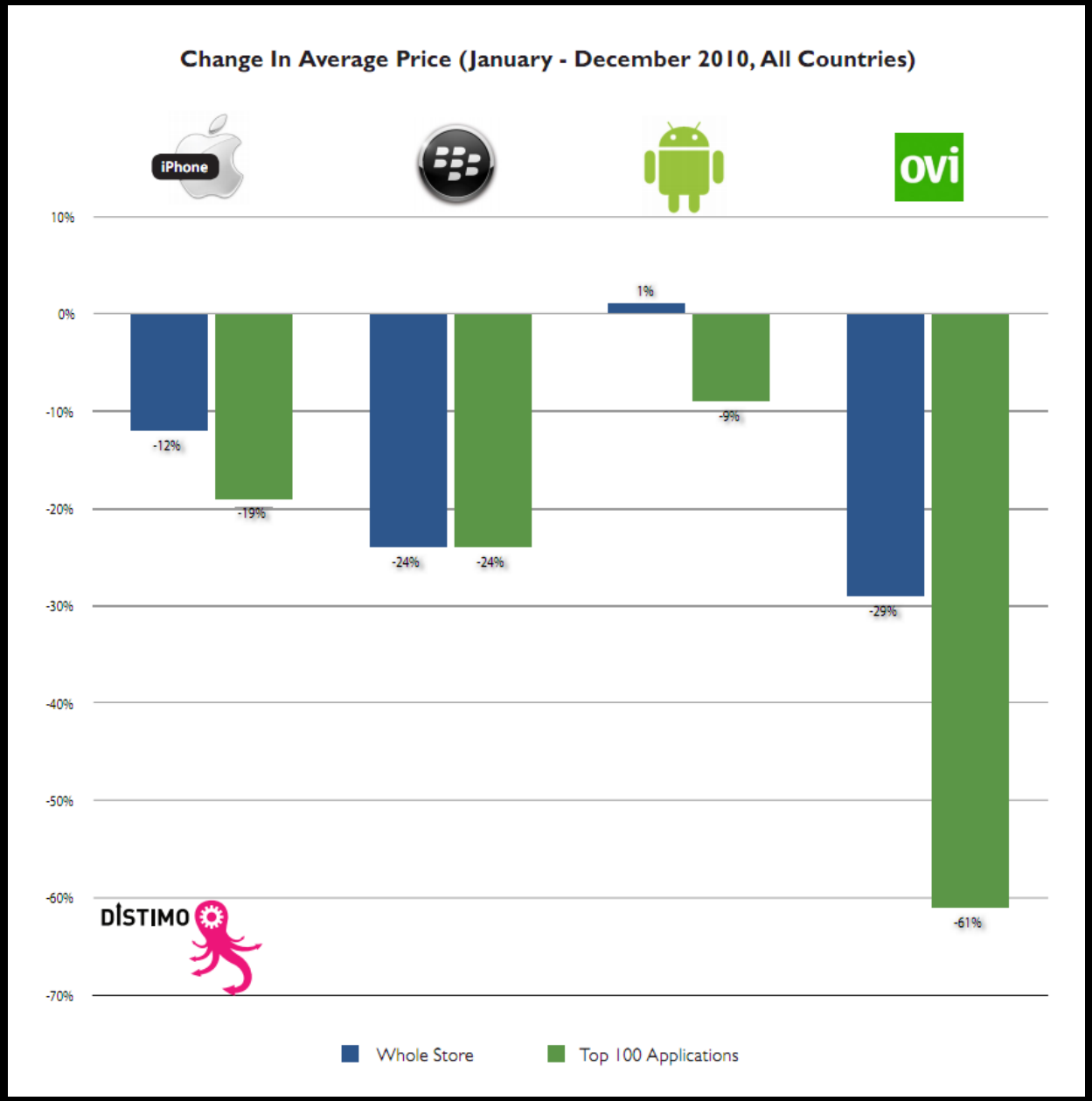
source: Distomo

Price Distribution All Stores (All Countries)

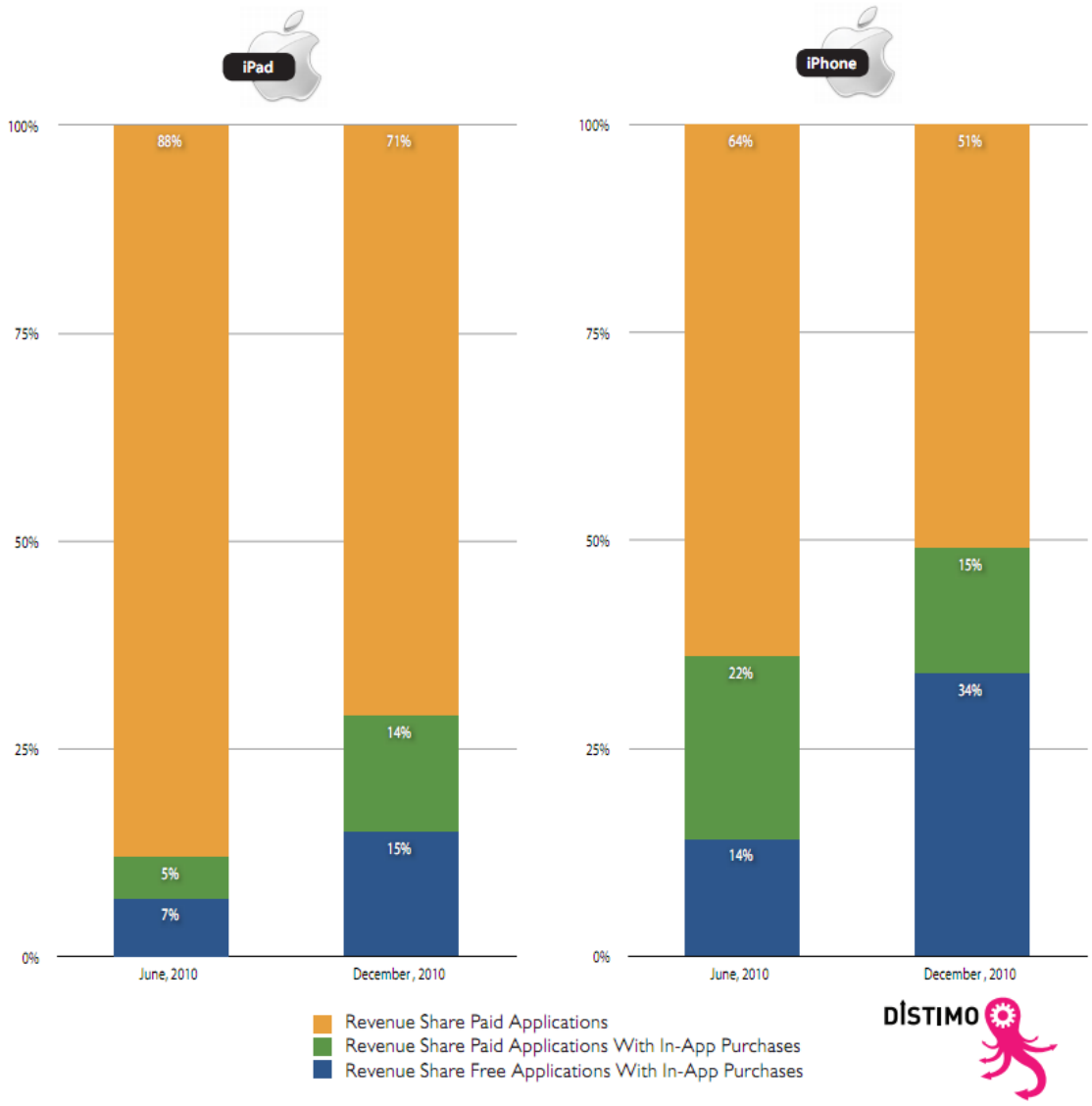




source: Distomo

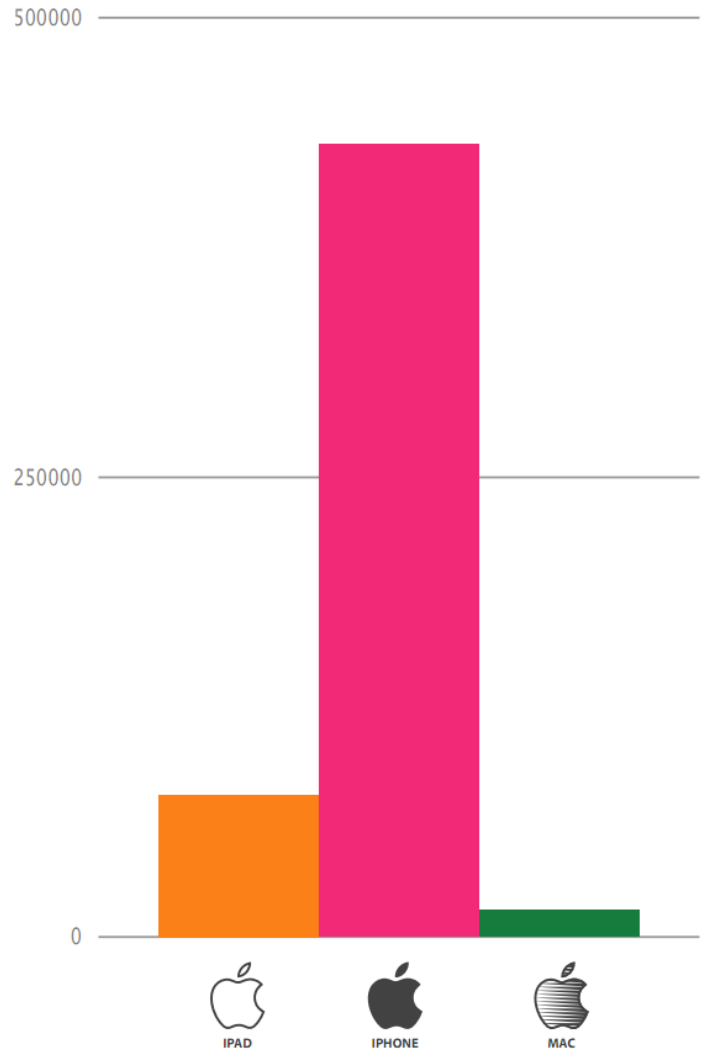

Change In Average Price (January - December 2010, All Countries)
iPhone
ovi
10%
0%
-10%
-20%
-30%
-40%
-50%
-60%
-70%
-12%
-19%
-24%
-24%
1%
-9%
-29%
-61%
DISTIMO
Whole Store
Top 100 Applications

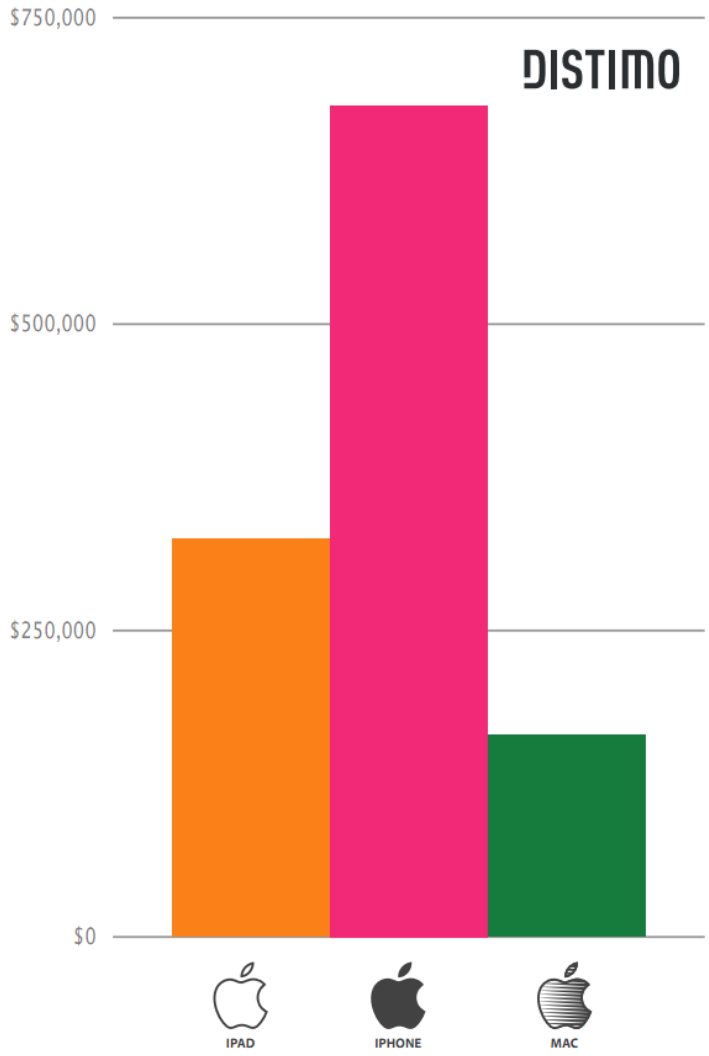
Revenue Share By Monetization Type (United States)



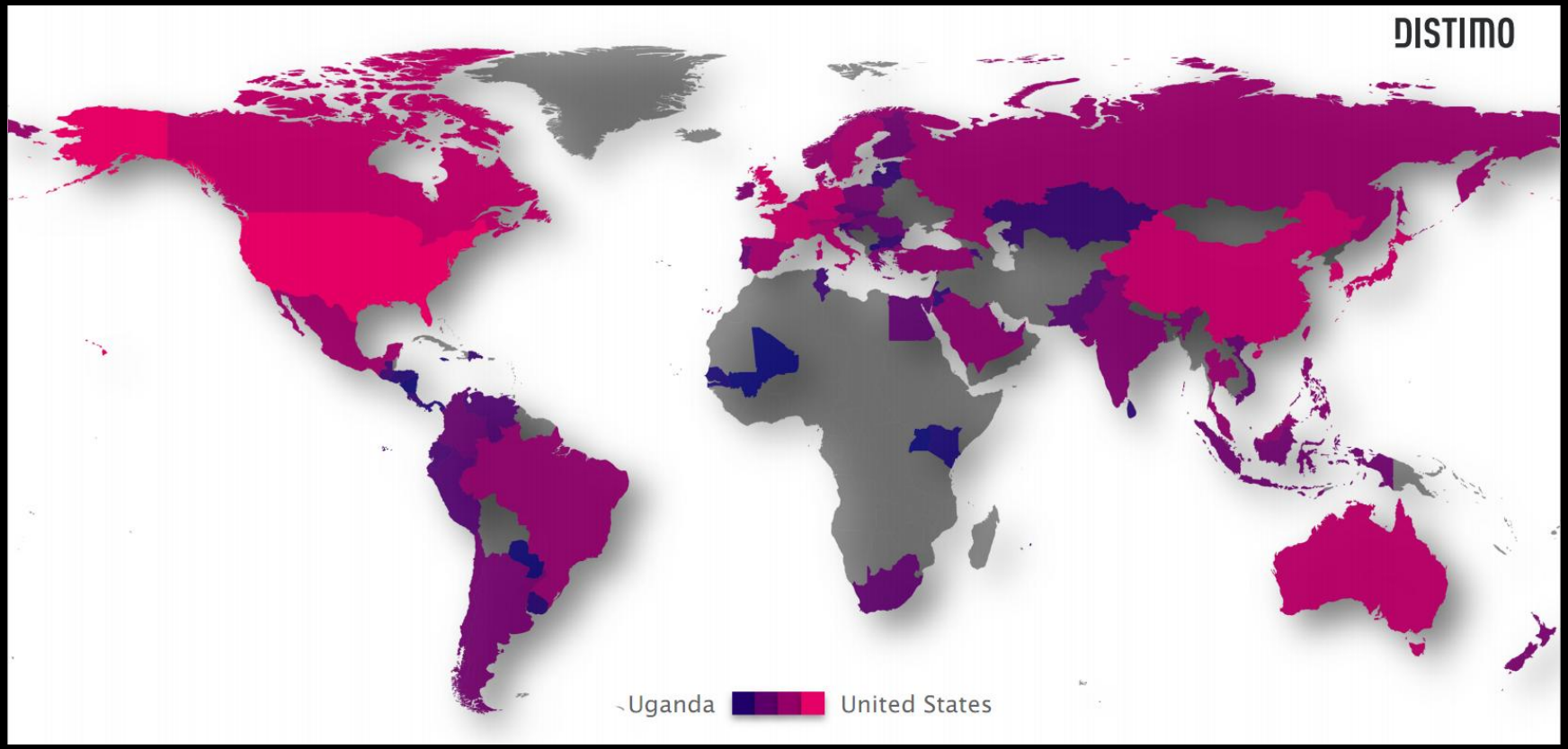
DOWNLOADS & REVENUE IN APPLE'S APPLICATION STORES

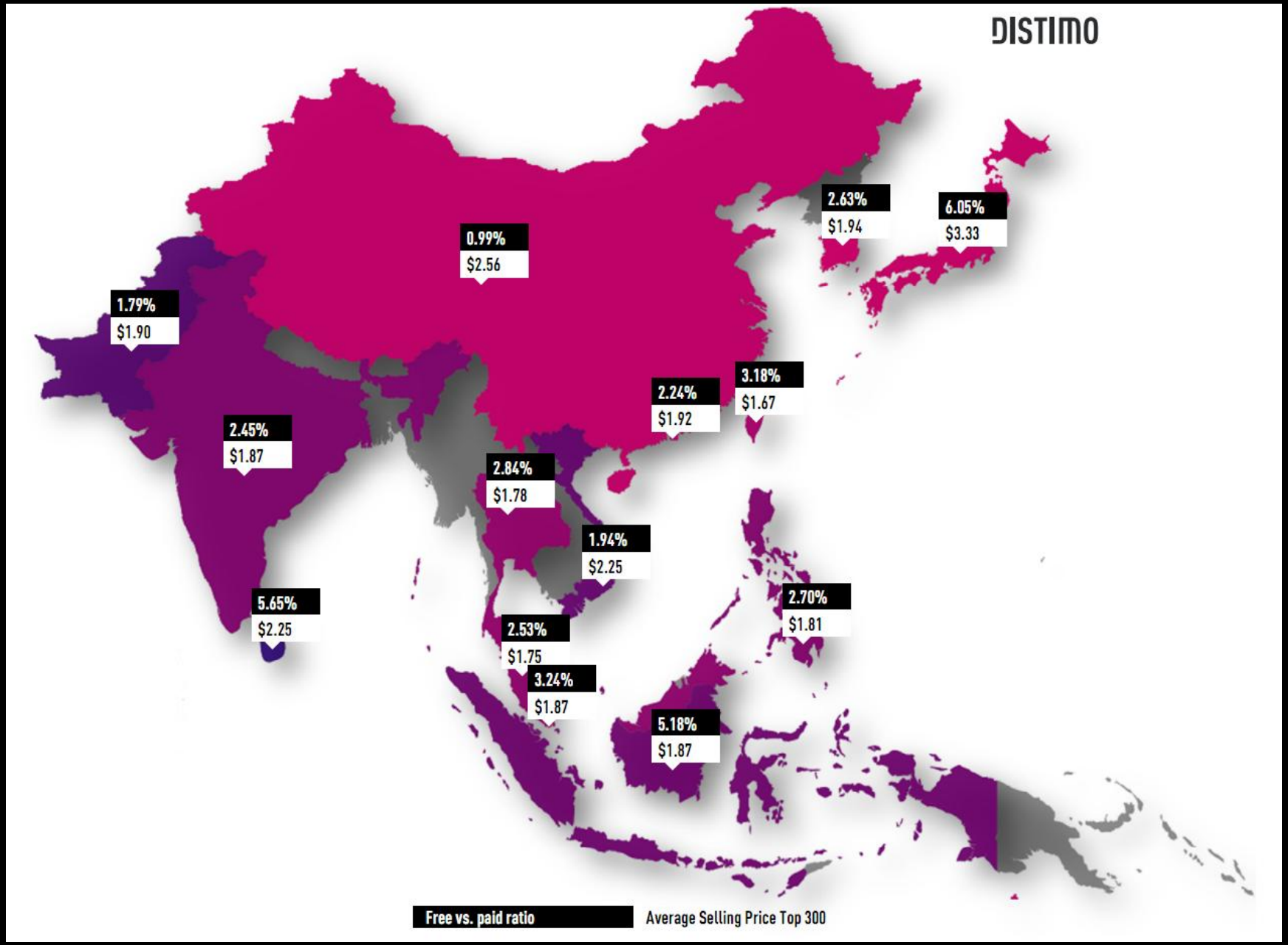
FEBRUARY 2011 - UNITED STATES

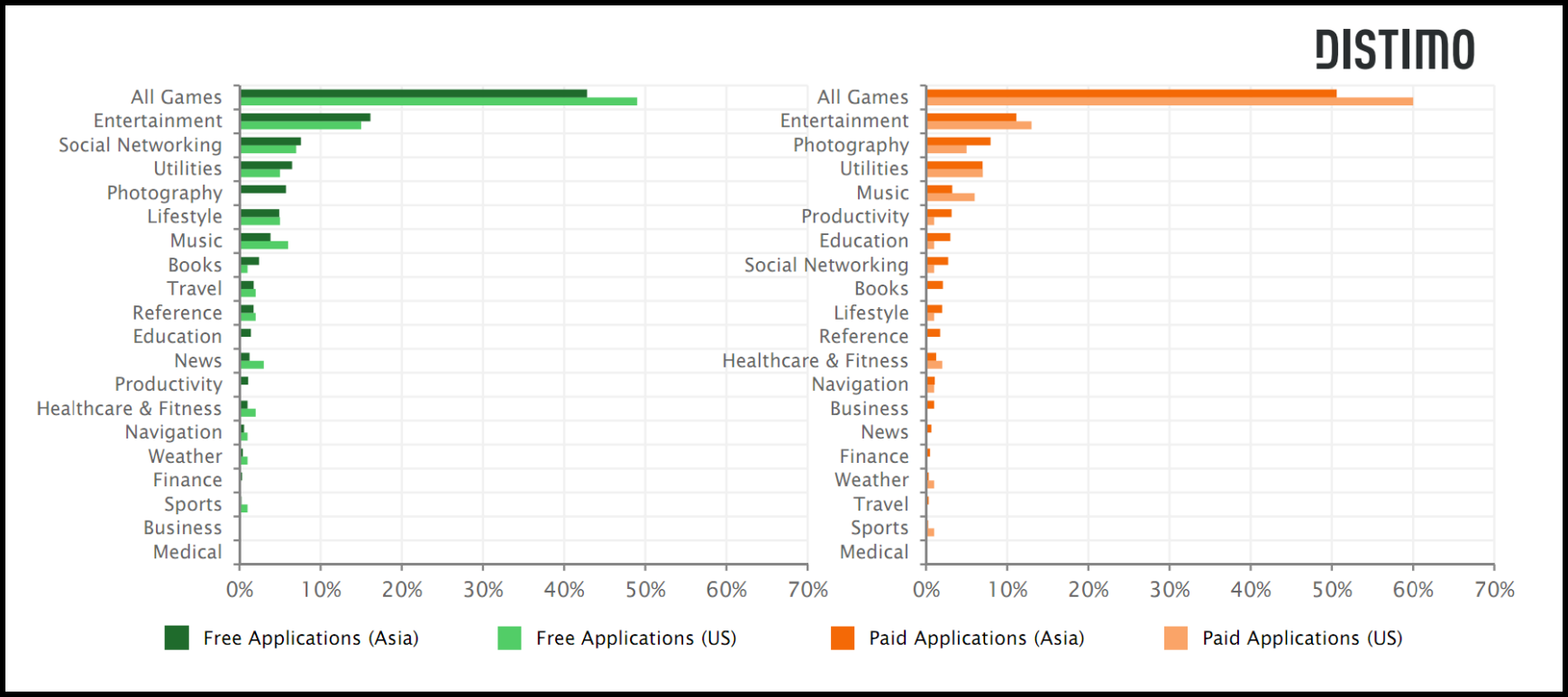




DISTIMO

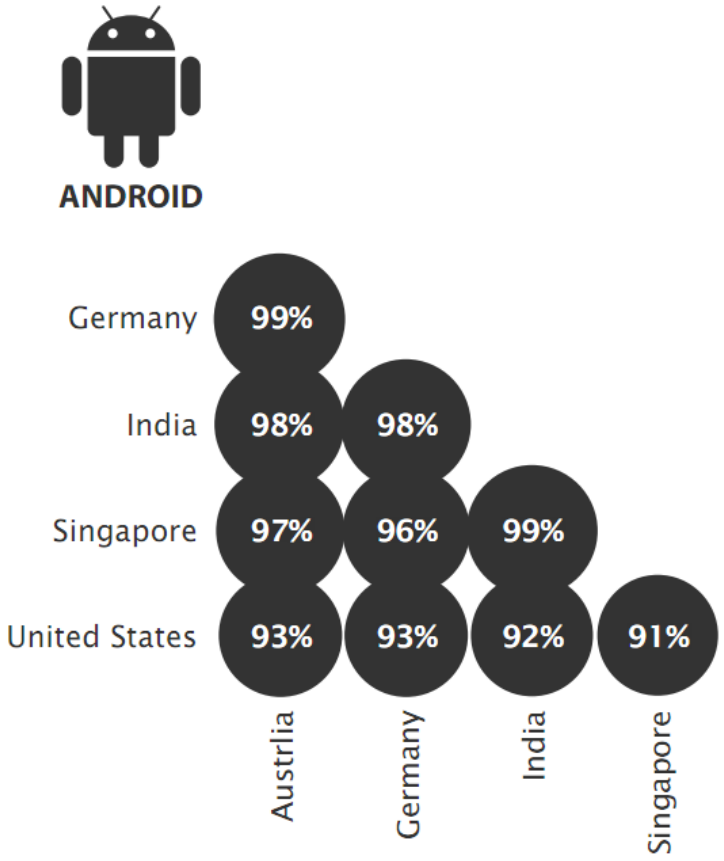

DISTIMO
Uganda
United States

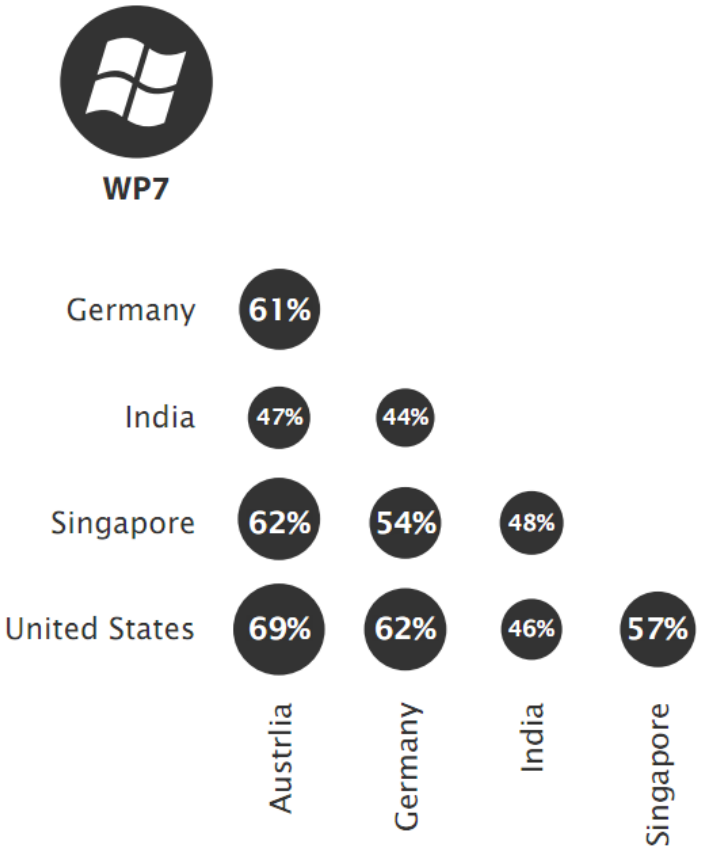

DISTIMO
0.99%
$2.56
2.63%
$1.94
6.05%
$3.33
1.79%
$1.90
3.18%
$1.67
2.24%
$1.92
2.45%
$1.87
2.84%
$1.78
1.94%
$2.25
5.65%
$2.25
2.70%
$1.81
2.53%
$1.75
3.24%
$1.87
5.18%
$1.87
Free vs. paid ratio
Average Selling Price Top 300


DISTIMO
All Games
Entertainment
Social Networking
Utilities
Photography
Lifestyle
Music
Books
Travel
Reference
Education
News
Productivity
Healthcare & Fitness
Navigation
Weather
Finance
Sports
Business
Medical
0%
10%
20%
30%
40%
50%
60%
70%
All Games
Entertainment
Photography
Utilities
Music
Productivity
Education
Social Networking
Books
Lifestyle
Reference
Healthcare & Fitness
Navigation
Business
News
Finance
Weather
Travel
Sports
Medical
0%
10%
20%
30%
40%
50%
60%
70%
Free Applications (Asia)
Free Applications (US)
Paid Applications (Asia)
Paid Applications (US)

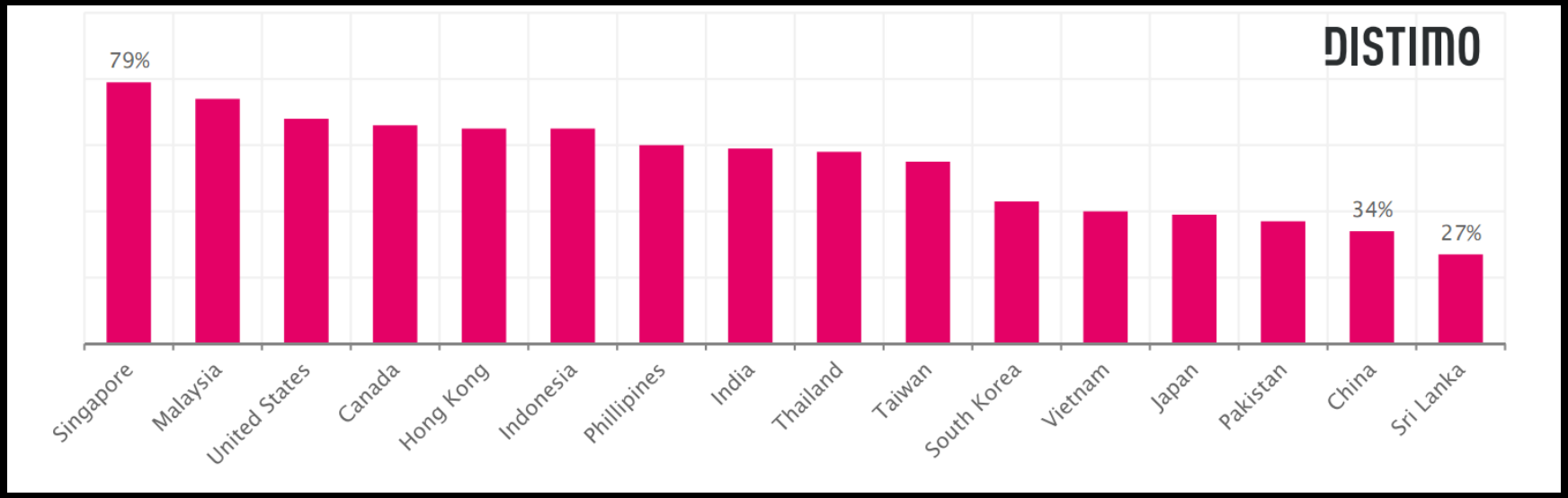
COUNTRY OVERLAP OF TOP APPLICATIONS

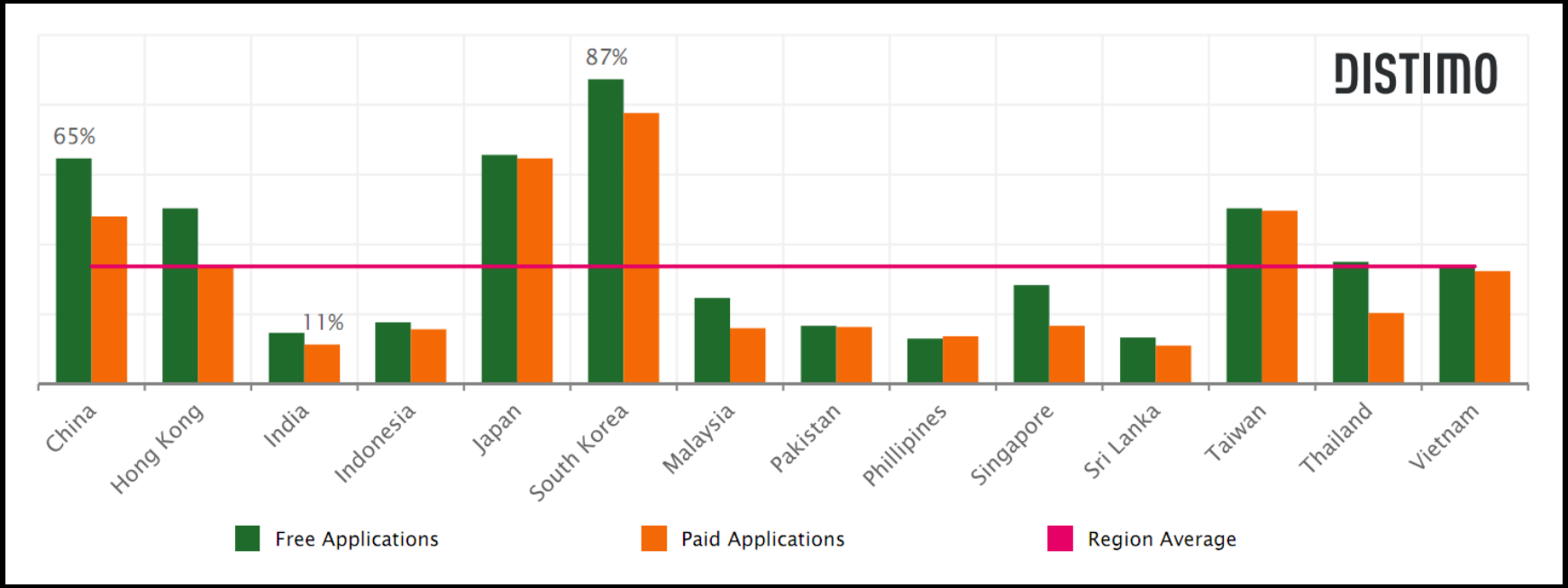
JANUARY 2011 - SELECTED COUNTRIES





source: Distomo


DISTIMO
79%
34%
27%
Singapore
Malaysia
United States
Canada
Hong Kong
Indonesia
Phillipines
India
Thailand
Taiwan
South Korea
Vietnam
Japan
Pakistan
China
Sri Lanka


DISTIMO
65%
11%
87%
China
Hong Kong
India
Indonesia
Japan
South Korea
Malaysia
Pakistan
Phillipines
Singapore
Sri Lanka
Taiwan
Thailand
Vietnam
Free Applications
Paid Applications
Region Average

mobiles vs. desktops

VS

VS

VS

vs

dev environments

iOS 4.3


Windows Phone
Mango Release 2011


webOS


BlackBerry®

iOS coding

Programming Mobile User Interfaces on Windows

Vidya Setlur

Principal Research Scientist

Nokia Research Center

Mobile Capabilities

- Rectangular bitmapped screens
- Text entry
- Audio and video support
- Network access
- Accelerated graphics hardware
- Smaller screen
- Touch-sensitive screen
- Motion Sensor
- Location Sensor

Red Threads

- Design principles for good user experience
  - Personal – Only display tasks involving the current user. Reduces the amount of data needed to be synchronized and displayed
  - Relevant – Inclusion of location service to filter data based on where user is located. Using accelerometer for movement, date and time information for temporal filtering
  - Connected – connect to online services, store information in cloud, device sharing

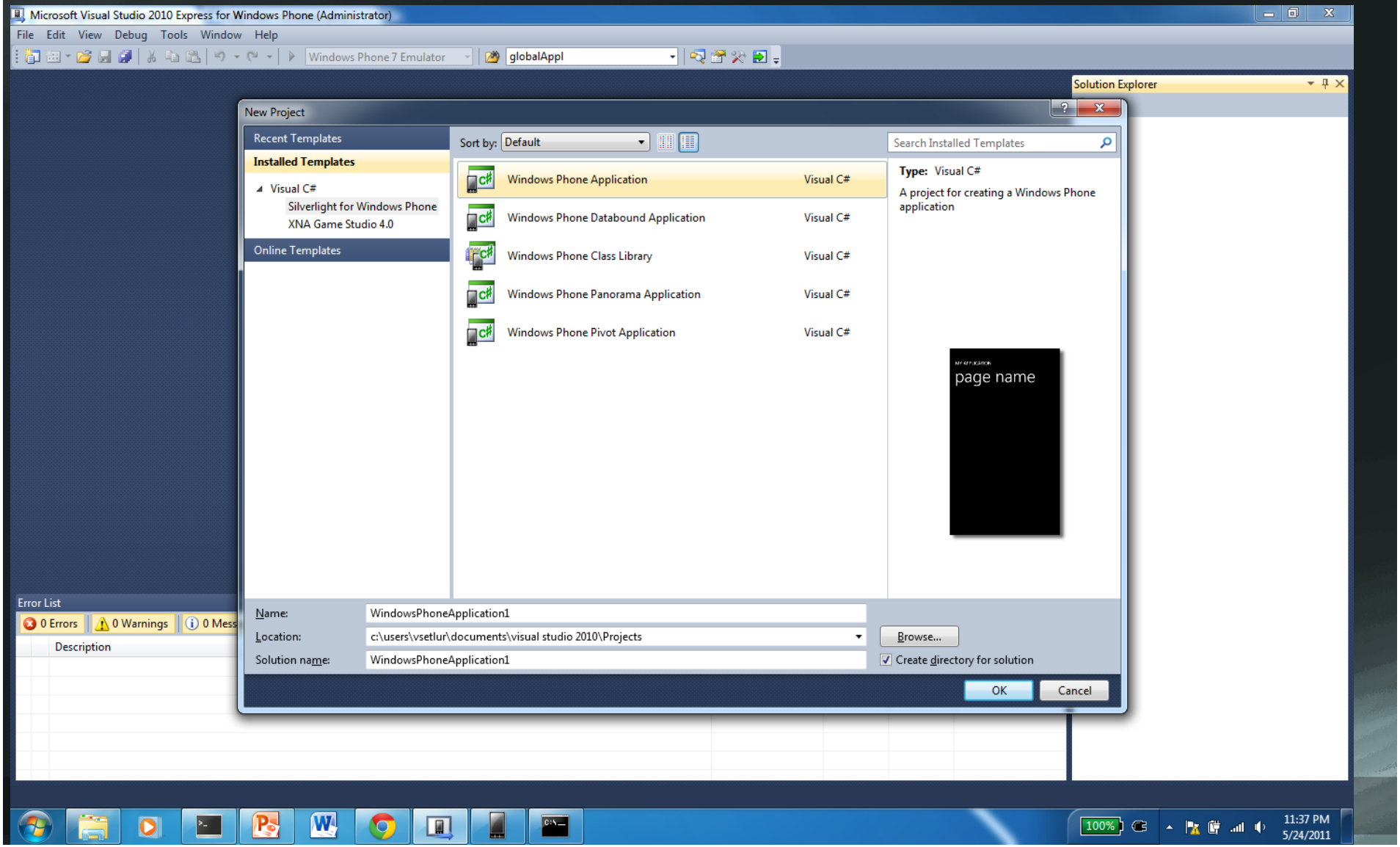
MVC Architecture

- Abstracts out a lot of the UI framework plumbing
- Supports a command framework used for event handling
- Abstracts platform-level details related to thread dispatch, background processing, etc.
- A portable way to do internationalization

Getting started

- http://create.msdn.com/en-US/
- Developer tools are:
  - Visual Studio Express for Windows Phone
  - Windows Phone Emulator
  - Silverlight for Windows Phone - (application development platform)
  - Expression Blend – (styles and animation)
  - XNA Game Studio

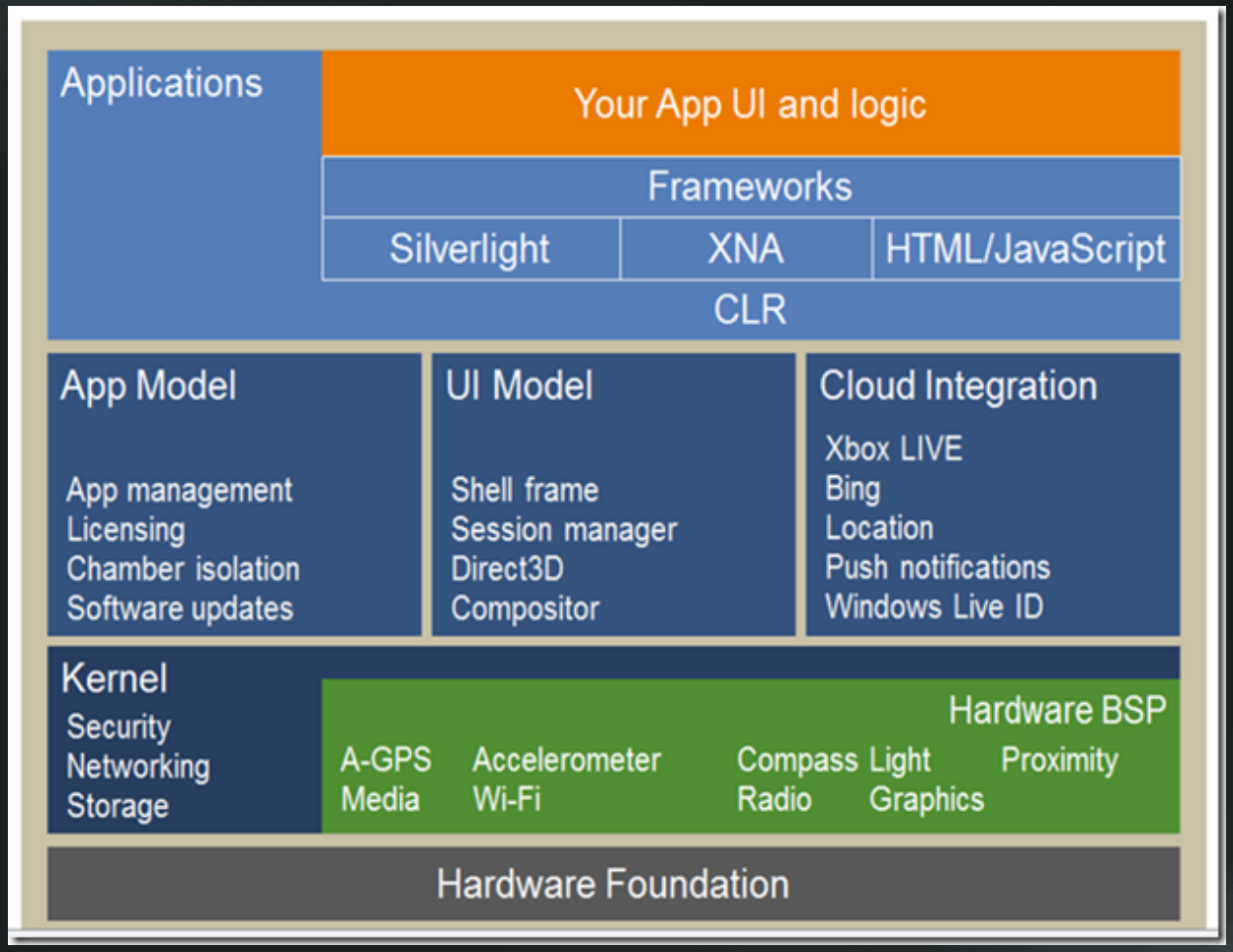
Getting started



Registering your device

- Make sure you have registered as a developer on the AppHub and have paid your subscription
- Install Zune software
- Plug in your phone via USB and let Windows setup the drivers
- Launch the Zune software and make sure that it picks up your device
- Unlock your device using the Windows Phone Developer Registration Tool which can be found in the Windows Phone Developer Tools folder of your Start Menu

WP7 Architecture



Hello World

- Launch Visual Studio and Select File -> New Project
- Select SilverLight for Windows Phone as template and Windows Phone Application as project type.
- MainPage.xaml: Default page with some UI element.
- App.xaml: Declares shared application resources like colors, brushes, fonts and various style objects.

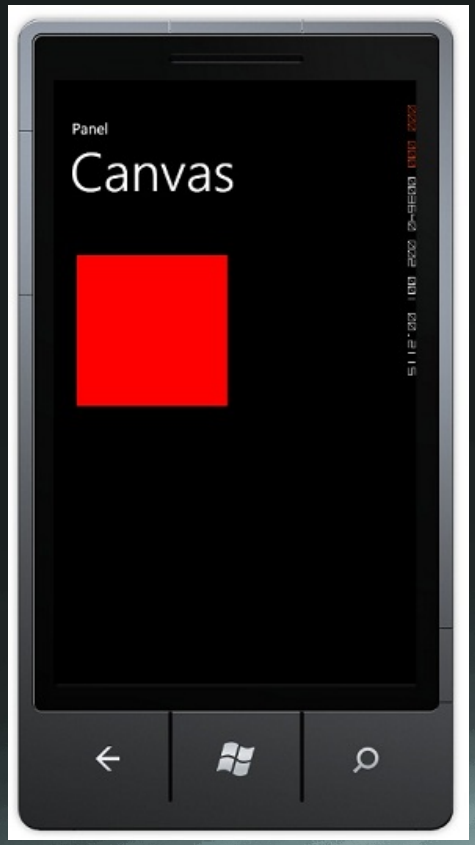
XAML and C#

- Every page has two files:
- XAML – eXtensible Avalon Markup Language. Defines layout
- C# - actual code
- However you can use them interchangeably

Layouts

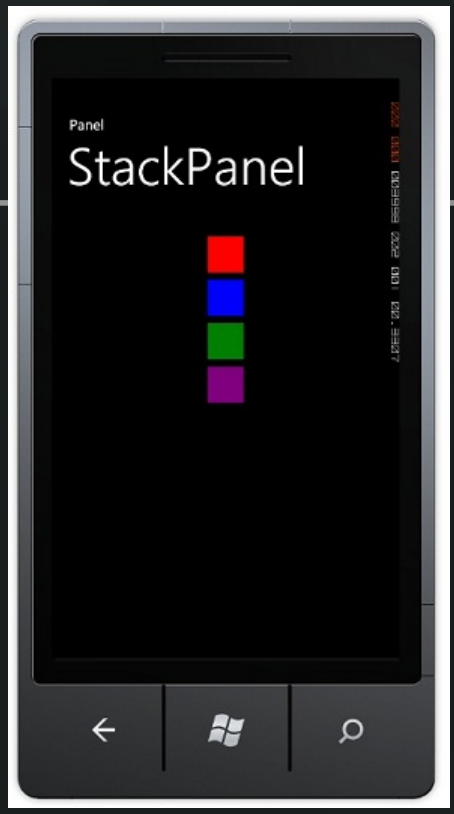
Canvas - It's typically used when children need to be positioned at exact x and y coordinates

```
<Canvas Background="Transparent">
<Rectangle Canvas.Left="30" Canvas.Top="200"
Fill="red" Width="200" Height="200" />
</Canvas>
```



Layouts

StackPanel - arranges content horizontally vertically (default)



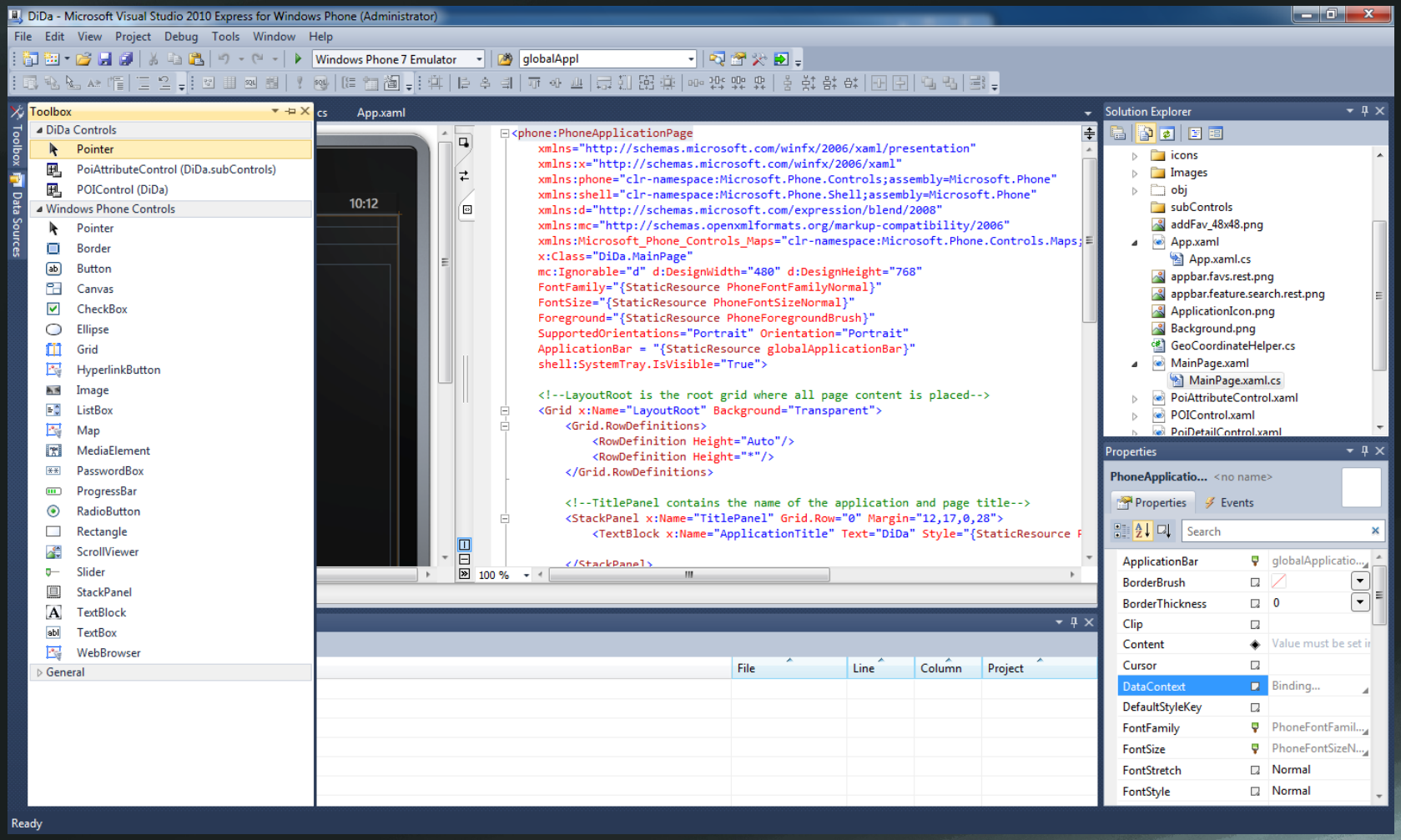

```
<StackPanel Margin="20">
    <Rectangle Fill="Red" Width="50" Height="50" Margin="5" />
    <Rectangle Fill="Blue" Width="50" Height="50" Margin="5" />
    <Rectangle Fill="Green" Width="50" Height="50" Margin="5" />
    <Rectangle Fill="Purple" Width="50" Height="50" Margin="5" />
</StackPanel>
```

Layouts

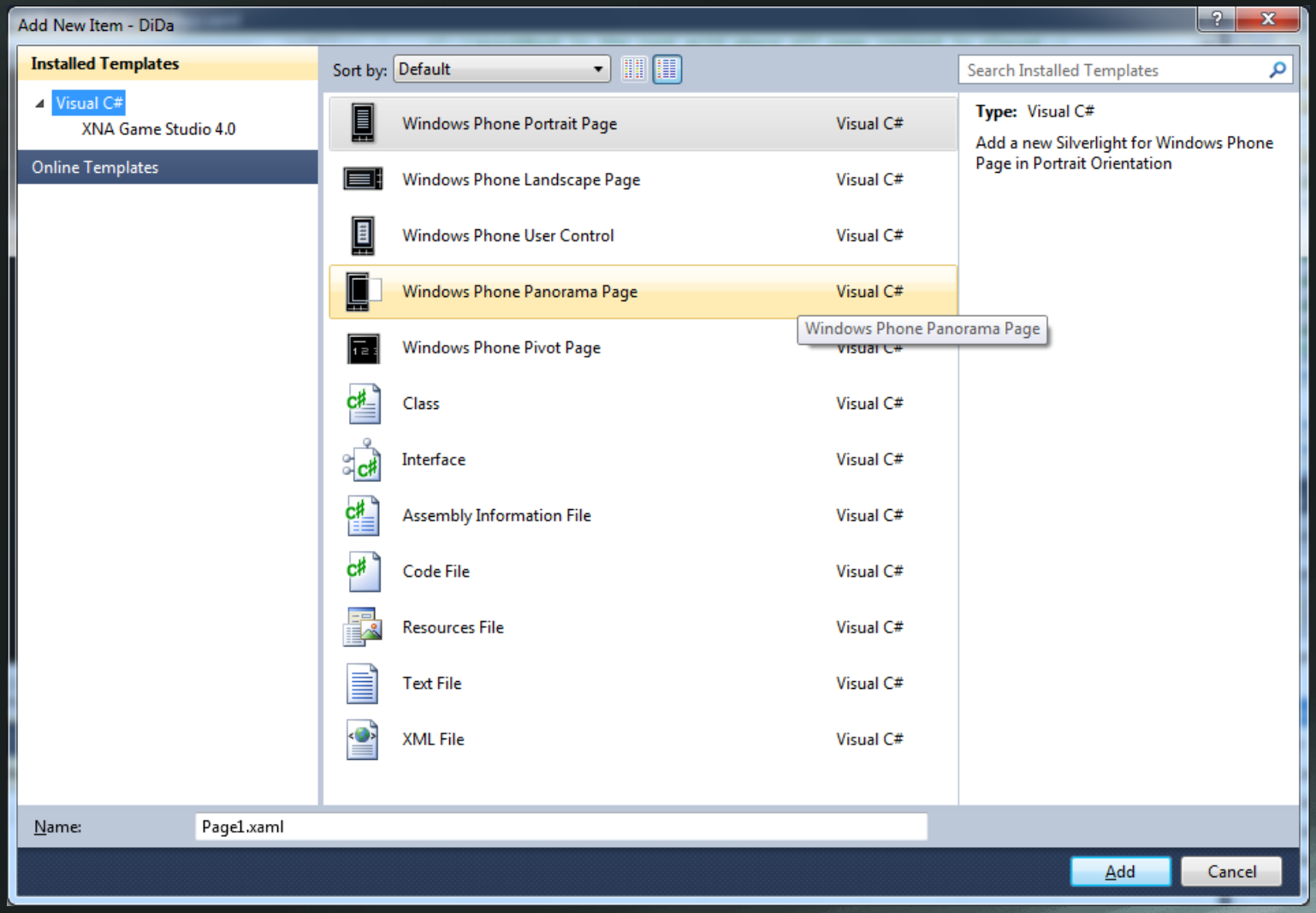
Grid – made up of rows and columns


```
<Grid>
  <Rectangle Fill="Red” Grid.Row="0” Grid.Column="0" />
  <Rectangle Fill="Orange” Grid.Row="0” Grid.Column="1" />
  <Rectangle Fill="Yellow” Grid.Row="0” Grid.Column="2" />
  ...
</Grid>
```

UI Controls


DiDa - Microsoft Visual Studio 2010 Express for Windows Phone (Administrator)
File Edit View Project Debug Tools Window Help
Windows Phone 7 Emulator
globalAppl
Toolbox
DiDa Controls
Pointer
PoiAttributeControl (DiDa.subControls)
POIControl (DiDa)
Windows Phone Controls
Pointer
Border
Button
Canvas
CheckBox
Ellipse
Grid
HyperlinkButton
Image
ListBox
Map
MediaElement
PasswordBox
ProgressBar
RadioButton
Rectangle
ScrollViewer
Slider
StackPanel
TextBlock
TextBox
WebBrowser
General
App.xaml
10:12
<phone:PhoneApplicationPage
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
x:Class="DiDa.MainPage"
mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
FontFamily="{StaticResource PhoneFontFamilyNormal}"
FontSize="{StaticResource PhoneFontSizeNormal}"
Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
ApplicationBar = "{StaticResource globalApplicationBar}"
shell:SystemTray.IsVisible="True">
<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent">
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="*"/>
</Grid.RowDefinitions>
<!--TitlePanel contains the name of the application and page title-->
<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
<TextBlock x:Name="ApplicationTitle" Text="DiDa" Style="{StaticResource
100 %
File
Line
Column
Project
Solution Explorer
icons
Images
obj
subControls
addFav_48x48.png
App.xaml
App.xaml.cs
appbar.favs.rest.png
appbar.feature.search.rest.png
ApplicationIcon.png
Background.png
GeoCoordinateHelper.cs
MainPage.xaml
MainPage.xaml.cs
PoiAttributeControl.xaml
POIControl.xaml
Properties
PhoneApplicatio... <no name>
Properties
Events
Search
ApplicationBar
globalApplicatio...
BorderBrush
BorderThickness
0
Clip
Content
Value must be set i
Cursor
DataContext
Binding...
DefaultStyleKey
FontFamily
PhoneFontFamil...
FontSize
PhoneFontSizeN...
FontStretch
Normal
FontStyle
Normal
Ready

Panorama and pivot controls

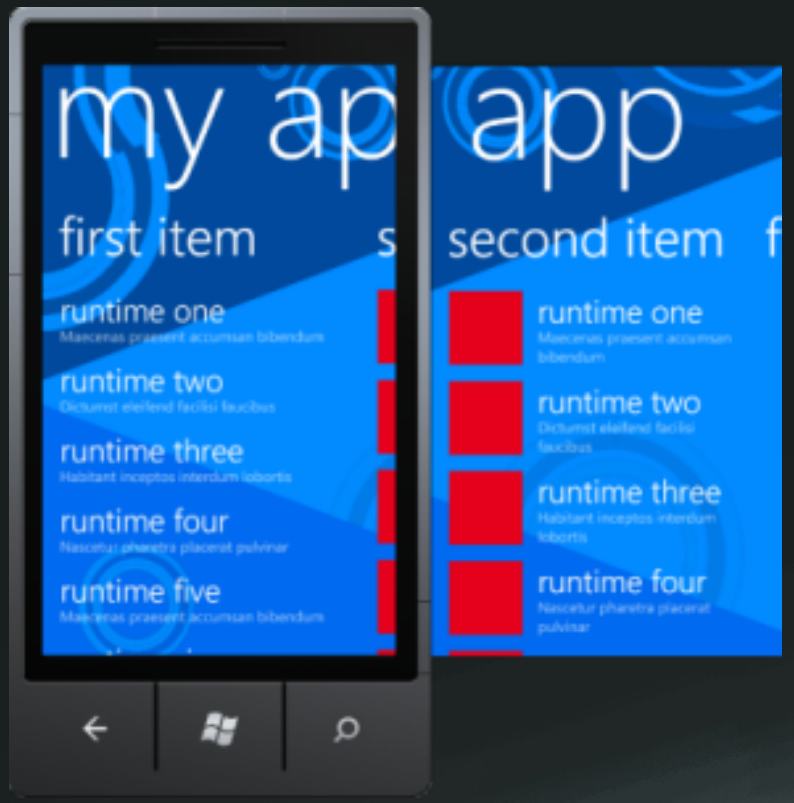


Panorama control

Gives the experience of a continuous view over several pages

On the right border of a page you see the continuation to the next page and you can flick between pages back and forth.

All Panorama items are rendered initially when the Panorama control is loaded.



Pivot control

Gives the experience of a “tabbed view“, optimized for mobile devices.

The pages are like single tabs. Not all tabs are rendered at once, instead a tab is rendered when a neighbor tab is activated.

You can change between the tabs with a flick gesture or by clicking the title of the tab.

Resources and Styles

- Customize the look and feel of a control
- Only `StaticResource` available for WP7
- Define in resource dictionary

```
<Application.Resources>

   <Color x:Key="ButtonBackgroundColor">#FFFF5C5C</Color>

<SolidColorBrush x:Key="ButtonBackgroundBrush"
Color="{StaticResource ButtonBackgroundColor}" />

</Application.Resources>
```

Themes

Modify standard resources such as `PhoneBackgroundBrush`

```
<Color x:Key="PhoneTextBoxColor">#99485C31</Color>
<SolidColorBrush x:Key="PhoneTextBoxBrush"
Color="{StaticResource PhoneTextBoxColor}"/>
```

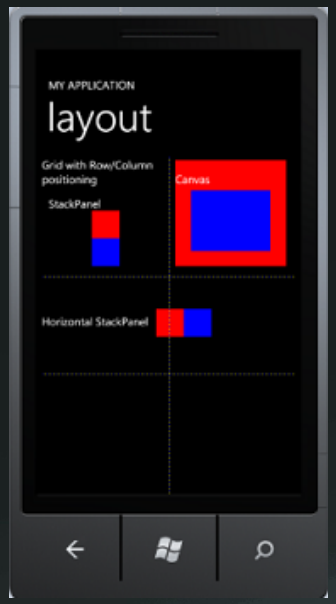
Themes – Best Practices

- For better performance compile your images with a "Build Action" of "Content" instead of the default "Resource" to reduce the size of your DLL, speeding up both app load and image load
- Wherever possible (if the image has no transparency) use JPEG images since these decode faster than PNG

Orientation

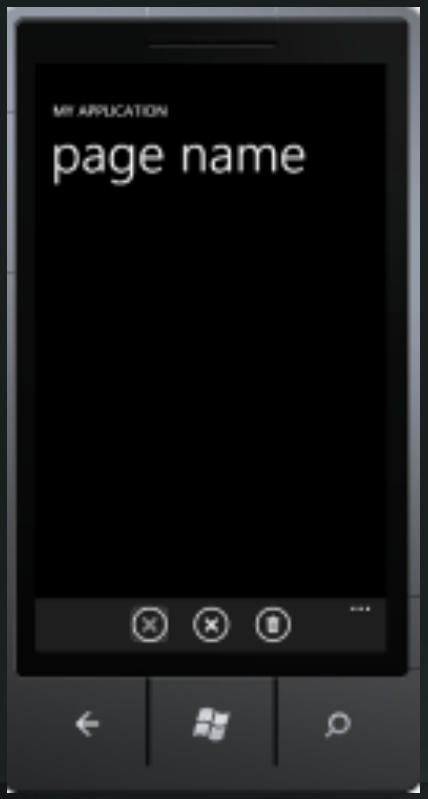
- Your application can detect device orientation by querying the `Orientation` property of the PhoneApplicationPage and using the `OrientationChanged` event handler

```
public enum PageOrientation
{
    None = 0,
    Portrait = 1,
    Landscape = 2,
    PortraitUp = 5,
    PortraitDown = 9,
    LandscapeLeft = 18,
    LandscapeRight = 34

}

var orientation = this.Orientation;
```



Application Bar

- Resides on the same side of the screen as the hardware buttons



Application Bar

```
<phoneNavigation:PhoneApplicationPage.ApplicationBar>

    <shell:ApplicationBar Visible="True" IsMenuEnabled="True">
    <shell:ApplicationBar.Buttons>
        <shell:ApplicationBarIconButton x:Name="btn1" IconUri="/
    Images/light/appbar.minus.rest.png" Click="btnUndo_Click" />

        ...

    </shell:ApplicationBar.Buttons>

    <shell:ApplicationBar.MenuItems>
        <shell:ApplicationBarMenuItem x:Name="mnuItemX"
        Text="Menu Item X" />

    </shell:ApplicationBar>

</phoneNavigation:PhoneApplicationPage.ApplicationBar>
```

Application Bar – UI Guidelines

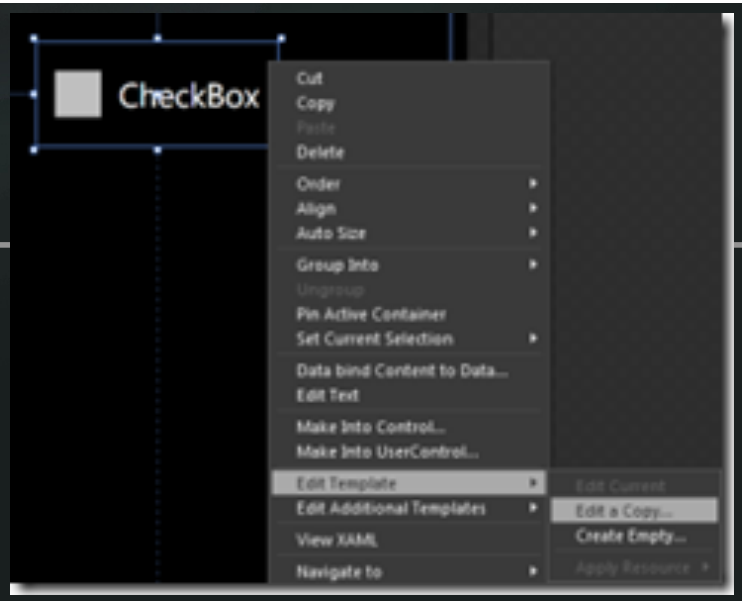
- Use the application bar button for common application tasks.
- You are limited to four application bar buttons.
- Place less frequently performed actions in the application bar menu.
- If the action is more complex, place it in the application bar menu instead of as a button.
- You are limited to five application bar menu items to prevent scrolling.
- Standard application bar icons (48x48) are installed as part of the Windows Phone Developer tools.

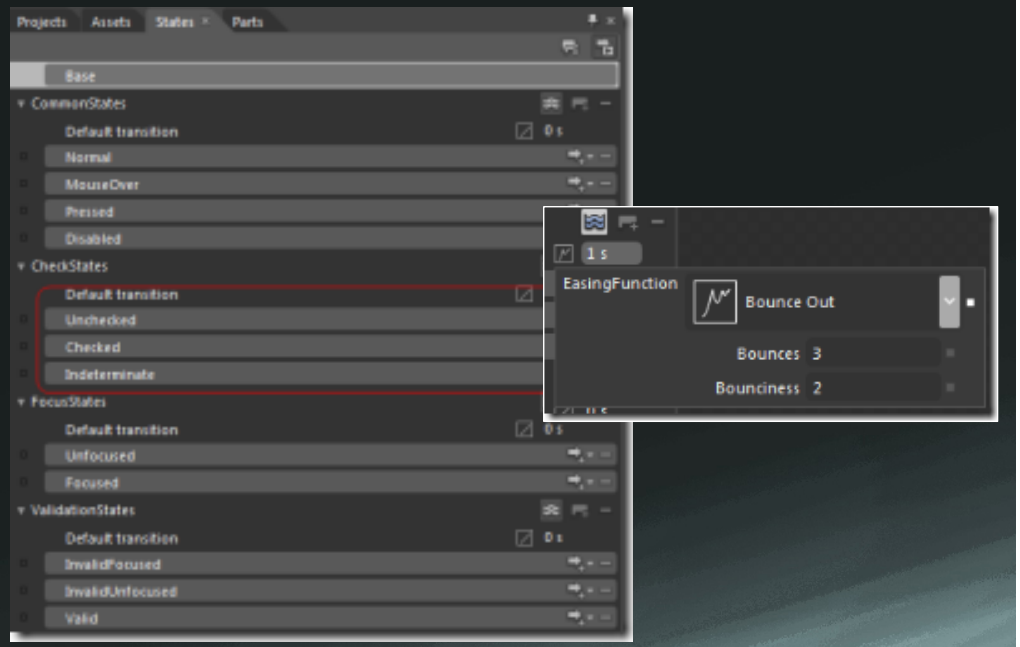
Visual States

- Specifies the visual behavior of the control based on certain states.
- VisualStateGroup objects are added to the VisualStateManager.
- VisualStateGroups represent states of a control.
- Switching between states is possible by calling the GoToState method.
- You put states that are mutually exclusive to each other in the same VisualStateGroup

Visual States

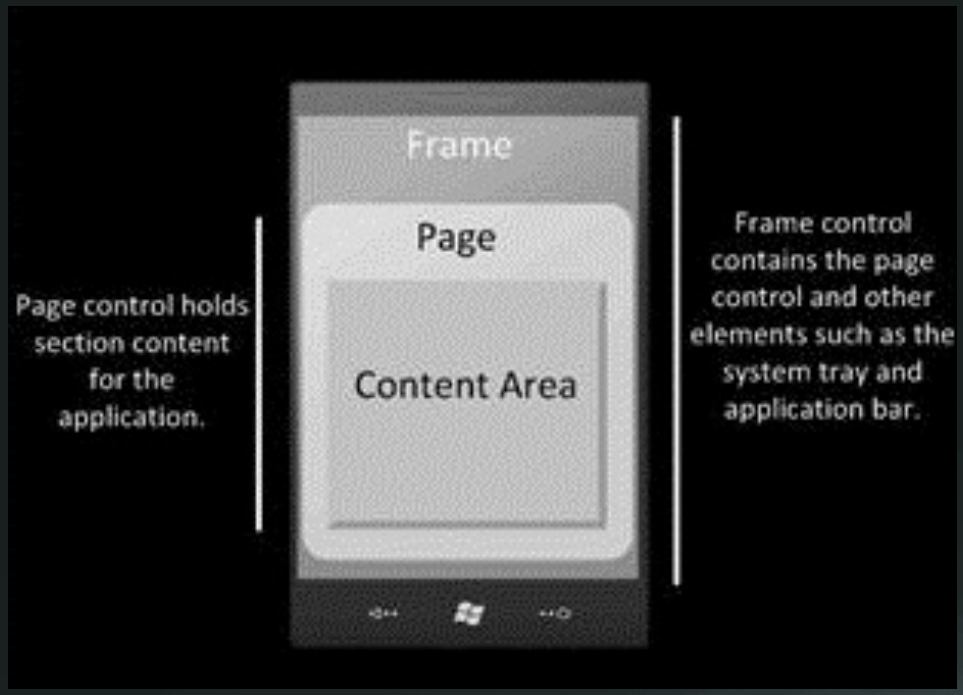
- You can use either Expression Blend or Visual Studio to change visual states.
- Blend is used more by designers than developers. Generates a lot of unnecessary code.
- Most developers use Blend only for a copy of the default state (default control template) and after that change the Visual State behaviors in Visual Studio





Navigation

- Based on a Silverlight page model where users can navigate forward through different screens of content. You can move backward using the "back" hardware button.
- Easily create view-based applications that fit naturally into the Windows Phone navigation model
- Provide default transitions that match the Windows Phone look and feel



Frame-Page Model

Navigation

There are three main ways to navigate between pages:

1. The PhoneApplicationPage class supports `OnNavigatedTo`, `OnNavigatedFrom` and `OnNavigatingFrom` virtual methods for handling the result of navigation:

```
protected override void OnNavigatedFrom(NavigationEventArgs e)
{
    base.OnNavigatedFrom(e);
  // some code here
}
```

Navigation

2. HyperlinkButton - provides `NavigateUri` property which can be used to navigate to a page Uri.

```
<HyperlinkButton Content="HyperlinkButton"
NavigateUri="/Page1.xaml"/>
```

3. NavigationService – more flexibility for navigation

`GoBack()`

`GoForward()`

`Navigate(Uri)` - Navigates to the content specified by the uniform resource identifier (URI).

Touch Events

- Tap
- Double-tap
- Pan
- Flick
- Touch and Hold
- Pinch and Stretch (Multi Touch)

Touch Events – UI Guidelines

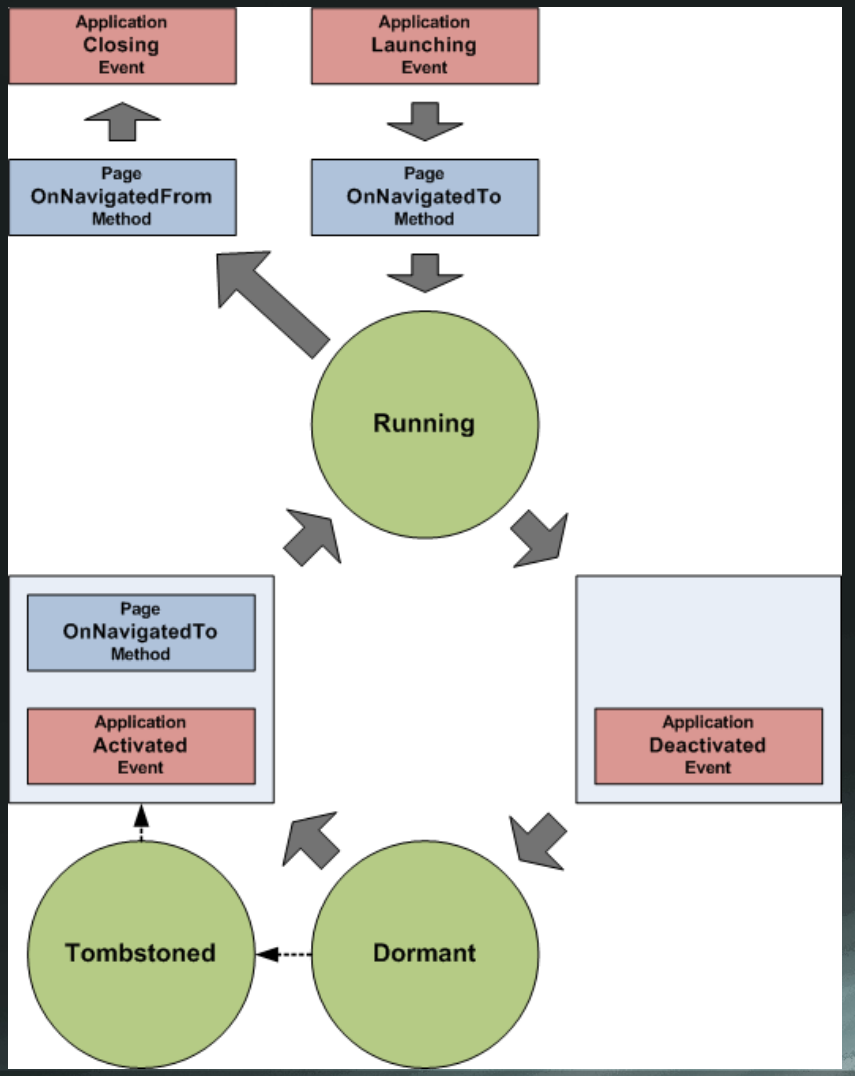
- Target size – Minimum dimension of a visual element that will respond to touch input is 7 mm (26 pixels)
- Layout – High frequency controls should be larger and positioned towards the middle of the screen, where they are easier to tap
- Gestures – Maintain the inherent semantics

Multitasking

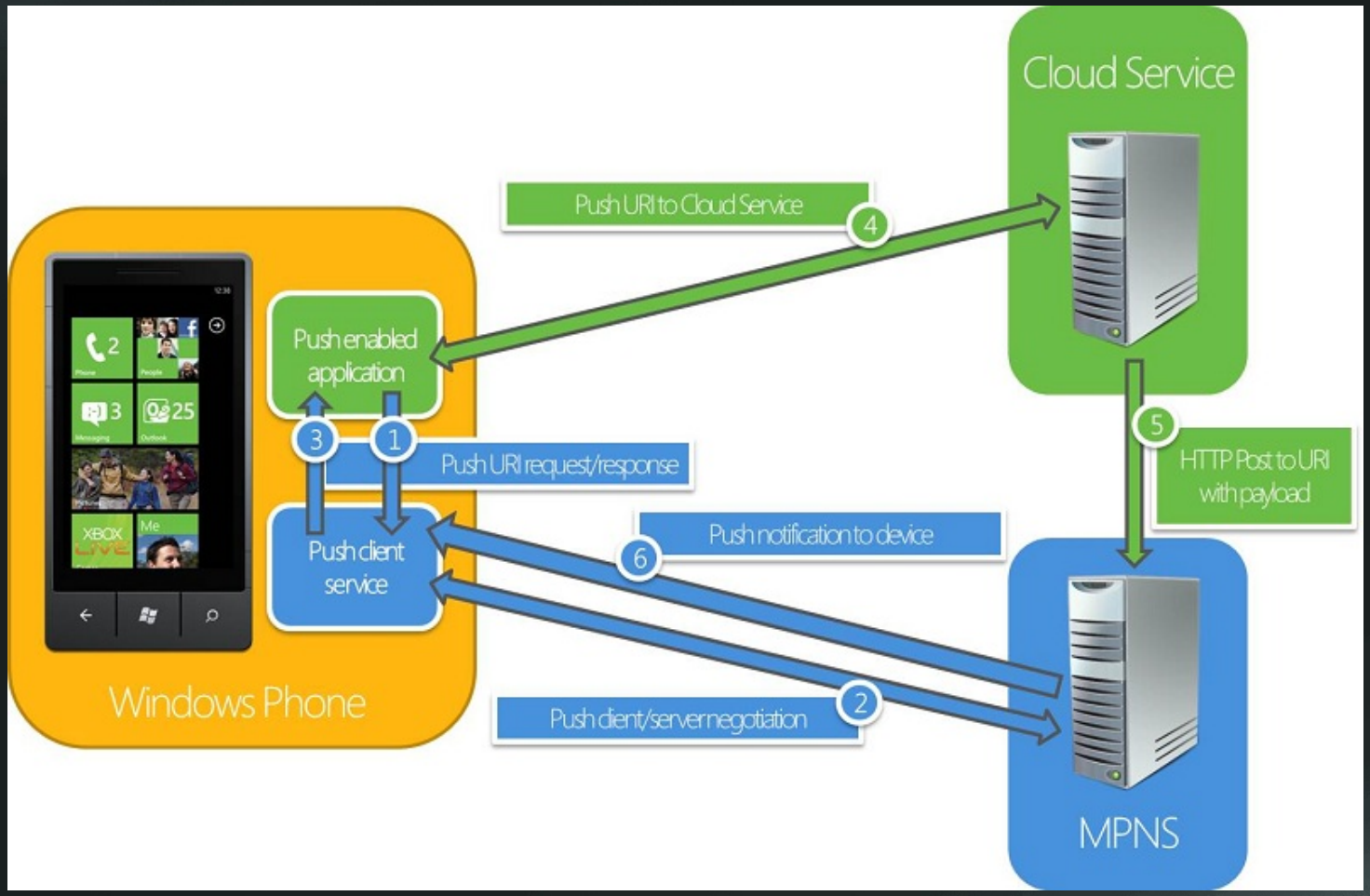
- Primary design considerations:
  - Prevent app from monopolizing CPU, power, memory
  - Promote a more continuous user experience
  - Ease programming with simple conceptual models
- Several mechanisms for multitasking:
  - Fast application resume framework
  - Push Notifications
  - Scheduled Tasks and Background agents

Execution Model

- Only one foreground app
- All others are dormant
- Low resources? *Tombstone* LRU apps
- Framework helps (re)store app state
- Also supports forward/back app navigation
  - Know your app's data and view state
  - Handle the 4 app state events
  - Handle page navigation events
- Store persistent state in Dictionary
- Save persistent data to a file on close



Notifications



Tasks and Background Agents

- ScheduledTask runs periodically in the background
- BackgroundAgent is invoked periodically by the task
- All tasks limited to: certain APIs, memory, and expiration time
- PeriodicTask: quick, frequently occurring (ex: upload)
  - Runs every 30 min (+/- 10 min) for at most 15 seconds
  - May not run at all if battery is low
  - No more than 6 periodic tasks on a device

Tasks and Background Agents

- ResourceIntensiveTask: long-running (ex: data sync)
  - Runs <= 10 min when deemed appropriate (power, CPU, memory)
  - Must be plugged-in, battery over 90%, and WiFi-connected
  - Screen must be locked, no active call

Other Multitasking

- Background Audio
  - Your app can play music and control volume
  - Other apps can run while this happens
- Background File Transfers
  - Your app can *download* in background
  - File size limit: 20 MB

Location API – Code

Main class: `GeoCoordinateWatcher`

1) Add Location Service DLL

2) Declare as a class variable

3) Set accuracy and movement threshold

4) Add event handlers

5) Start() and Stop() the watcher

Event handlers:

`StatusChanged` - Meta-level info (e.g., initializing, ready)

`PositionChanged` -New location information

Location API – UI Guidelines

- Conserving power - Best accuracy at lowest power using GPS, GSM, and WiFi
- How to specify level of accuracy ?
  - Constructor parameter sets coarse level (i.e., default, high)
  - Property specifies "movement threshold" to limit updates
- How is power saved ?
  - Use GPS only for high accuracy, o/w depend on WiFi/GSM
  - Turn off location service when not in use !

Using Bing Maps

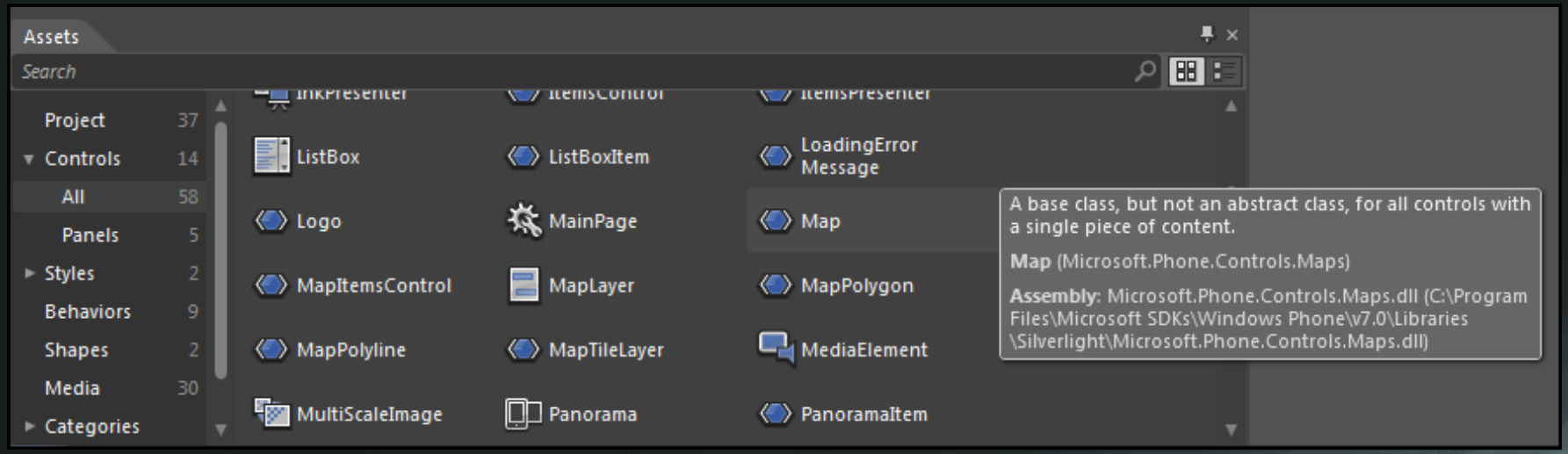
- Map Design
- Credentials
- Pushpins
- Events
- Web Services

Using Bing Maps

First, add a reference Microsoft.Phone.Controls.Maps.dll to your project.

Two ways of adding a Map Control

- Project - > Open in Expression Blend, Select the xaml file



- Or programmatically

Using Bing Maps

Need an application key. Sign up at www.bingsmapsportal.com

Or an ugly message is displayed across the map indicating no valid credentials provided.

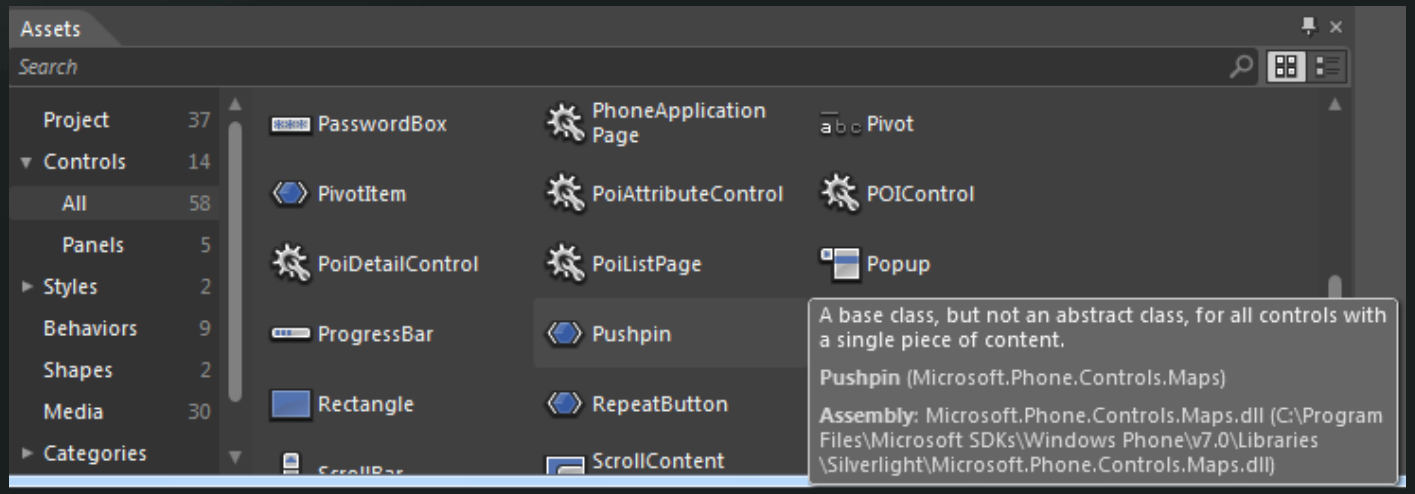
In XAML: use the CredentialsProvider attribute

In C#:

```
map.CredentialsProvider = new
ApplicationIdCredentialsProvider("<credentials");
```

Pushpins

- Either use Blend-> Assets to drag a pushpin onto the map



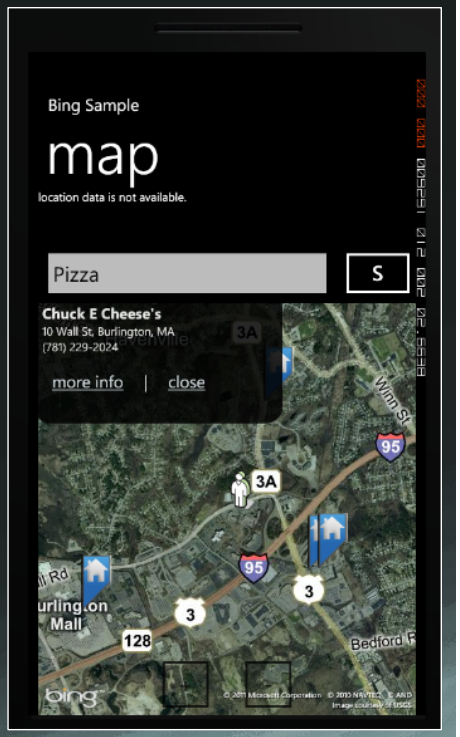
- Or in C#

```
Pushpin pin = new Pushpin();

pin.Location = new GeoCoordinate(lat, lon);

map.Children.Add(pin);
```

Customize pushpins

Use application resources. The application definition file (App.xaml) defines a resource section. Resources defined at the application level can be accessed by all other pages that are part of the application.

```
<Application.Resources>

     <Style TargetType="Button" x:Key="GelButton" >
     <Setter Property="Margin" Value="1,2,1,2"/>

     ...

     </Style>

</Application.Resources>
```



Map Events

- ViewChangeStart, ViewChangeOnFrame and ViewChangeEnd on the Map Control
- MousePress on the Pushpin Control

Bings Maps Web Services

- Geocode service – converts geographic entities to latitude, as well as return location information for a specified latitude and longitude.
- Route service - generates routes and driving directions based on locations or waypoints
- Imagery service - retrieves information about imagery data as well as getting URIs for maps. For example, get a link to a map with a pushpin at a specific location or provide a road map or bird's eye imagery
- Search service - parses a search query that contains a location or keyword (or both) and return search result.

Accelerometer

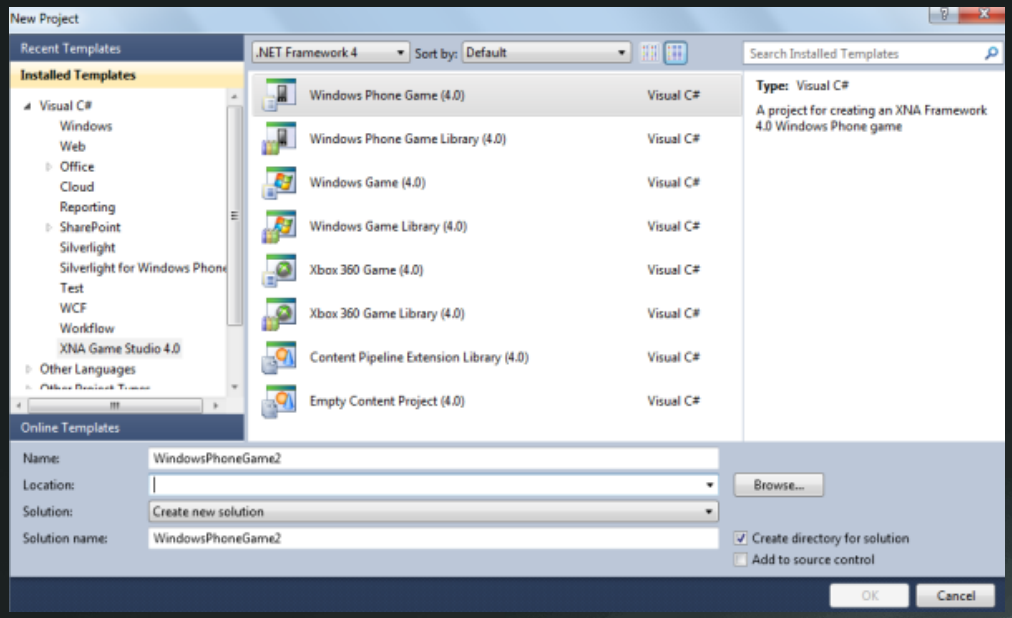
- Basic 3-axis accelerometer: x, y, z between -2.0 to 2.0
- Similar usage model to Location service
  - Start and stop events
  - New readings reported through events
- A few differences:
  - Update rate can be set (as timeout between updates)
  - Handlers run in background thread (delegate to update UI)

Other Sensors

- Camera
  - Activate shutter, auto focus, res, flash, etc.
  - Capture and alter live ARGB video frames
- Microphone
  - Record/playback audio with Microphone
  - Add Microsoft.Xna.Framework
  - See “Silverlight Microphone Example”

3D Graphics

- Uses XNA framework, based on DirectX
- Video card with DirectX 10.1 support needed to run emulator.
- Install the latest DirectX SDK before installing the Windows Phone Developer Tools.
- After installation has completed, create a new Windows Phone Game (4.0) project in Visual Studio.



XNA Framework

- The XNA Framework contains elements like math library for working with vectors and matrixes, a library for unified work with different input controllers, etc. .
- Application Model - is a framework (template) for an application.
- Content Pipeline - unifies game content processing. Content is processed with importers and processors

```
Model model;
protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
    model = Content.Load<Model>("teapot");
}
protected override void Draw(GameTime gameTime)
{
    // Clear frame buffer
    GraphicsDevice.Clear(Color.CornflowerBlue);
    Matrix world = Matrix.CreateRotationY((float)gameTime.TotalGameTime.TotalSeconds);
    Matrix view = Matrix.CreateLookAt(new Vector3(0,0,2), Vector3.Zero, Vector3.Up);
    Matrix projection = Matrix.CreatePerspectiveFieldOfView(
    MathHelper.ToRadians(45),
    GraphicsDevice.Viewport.AspectRatio, 0.1f, 10f);
    model.Draw(world, view, projection);
    base.Draw(gameTime);
}
```

Summarizing Wp7 UI Guidelines

- Mockup the pages and navigational map of your application and walk through them several times before coding. This will minimize or eliminate the need to add pages or change the map later, when it will be much harder
- Pressing the back button from the first screen of an application must exit the application
- Pressing the back button must return the application to the previous page
- Portrait is the default application view-you must add code to support landscape view

Summarizing Wp7 UI Guidelines

- Tile images should PNG format and measure 173 pixels by 173 pixels at 256 dpi
- Make sure to change Build Action for images to Content when you add them to Visual Studio
- Avoid using too much white in applications, such as white backgrounds, as this may have an impact on battery life for devices that have organic LED displays

Summarizing Wp7 UI Guidelines

- Application actions that overwrite or delete data, or are irreversible must have a "Cancel" button
- All basic or common tasks should be completed using a single tap gesture
- The touch and hold gesture should generally be used to display a context menu or options page for an item

Summarizing Wp7 UI Guidelines

- Canvas uses a pixel-based layout and can provide better layout performance than the grid control applications that do not change orientations
- Grid control is the best choice when the application frame needs to grow, shrink, or rotate
- Use a pivot control to filter large data sets, providing a view of multiple data sets, or to provide a way to switch between different views of the same data

Summarizing Wp7 UI Guidelines

- Use fonts other than Segoe sparingly
- Avoid using font sizes that are smaller than 15 points in size
- Maintain consistent capitalization practices to prevent a disjointed or jagged reading experience
- The title bar application title should be all capitals
- User all lower case letters for most other application text including page titles, list titles, etc.

Coding Exercise

- Derive your own class from BackgroundAgent
- Register as a periodic or resource intensive task
- It will be invoked with OnInvoke(ScheduledTask)
- Do your work with the APIs and resources you have
- If you finish then call NotifyComplete()
- If you don't complete then Abort()
  - If you abort then the OS won't execute your task again

Coding Exercise

- Main class: Accelerometer
- Produces: AccelerometerReading objects

  1) Add Microsoft.Devices.Sensors

  2) Declare Acclerometer as a class variable

  3) Set TimeBetweenUpdates (in ms)

  4) Add event handler

  5) Start() and Stop() the sensor

- One event to handle:

  - CurrentValueChanged: new x,y,z triple

3D Graphics and Cameras

Kari Pulli
Senior Director
NVIDIA Research

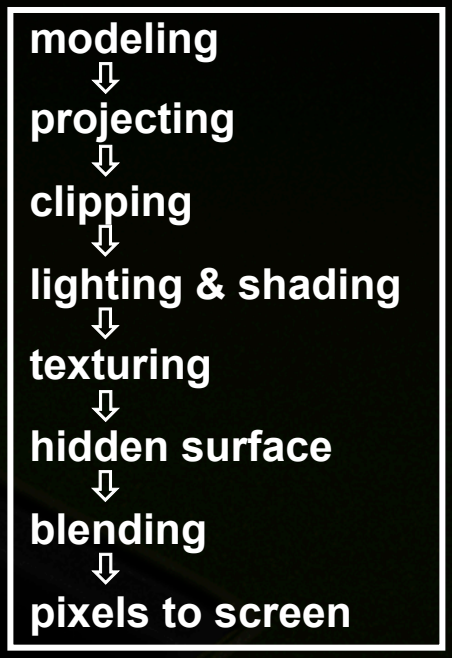
Overview

- OpenGL ES 1.x
- OpenGL ES 2.0
- WebGL
- OpenCV
- FCam

- All examples on this session on Android

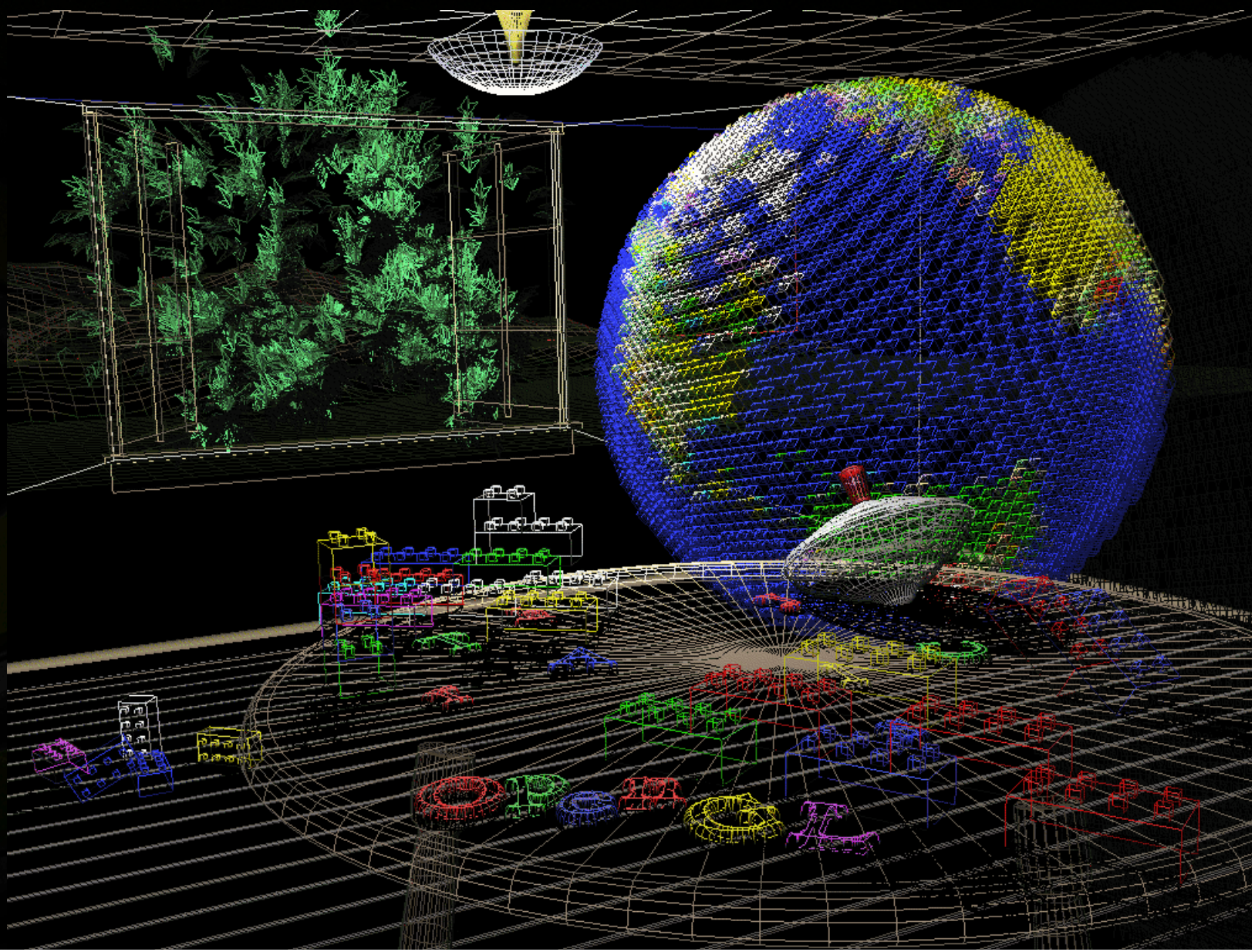
OpenGL

- The most widely adopted graphics standard
  - most OS's, thousands of applications
- Map the graphics process into a pipeline
  - matched traditional graphics HW well

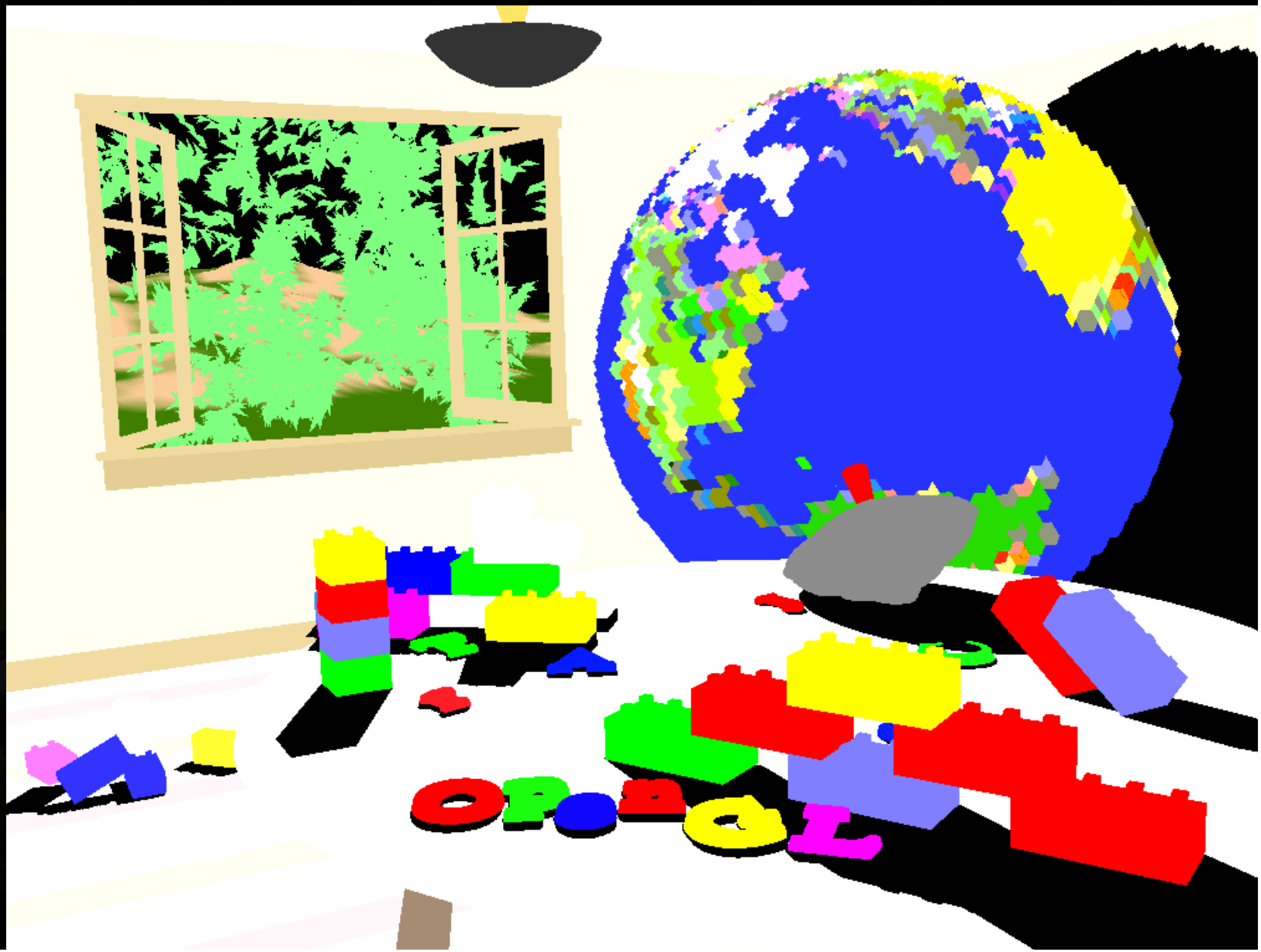


- A foundation for many higher level APIs
  - Open Inventor; VRML / X3D; Java3D; game engines

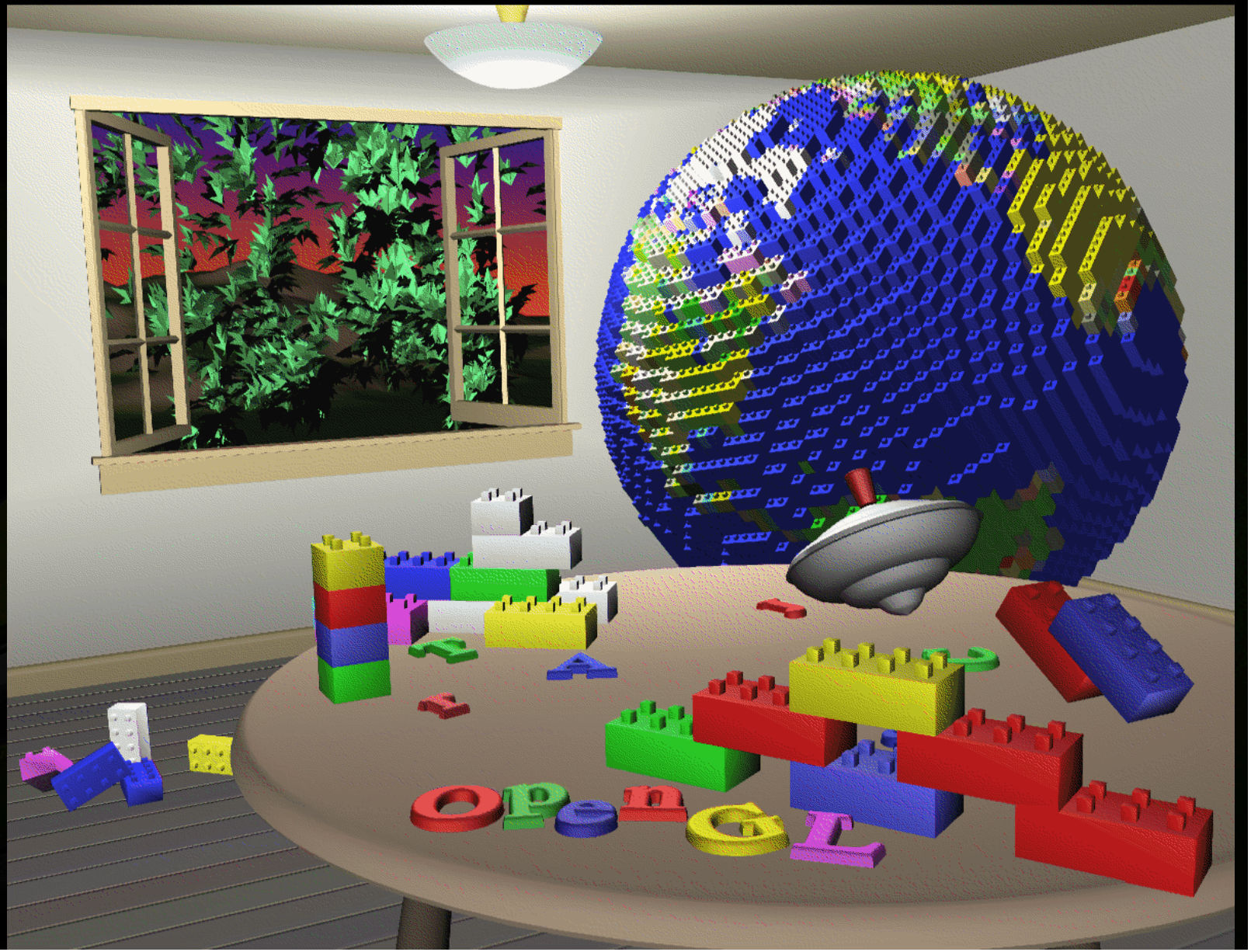
Project vertices to camera…



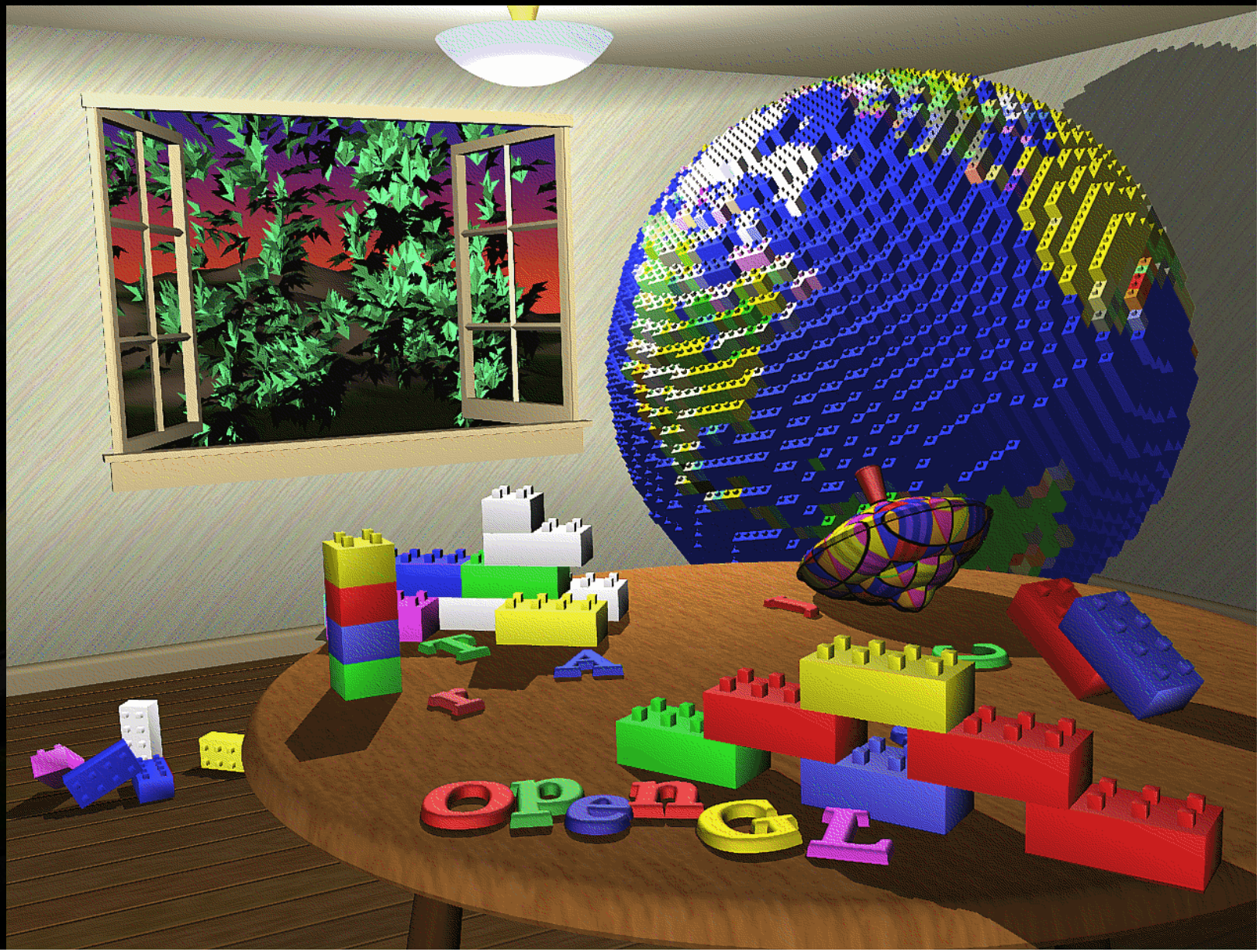
… connect them to triangles …



… shade them based on lights …



… add texture maps, shadows, …



What is OpenGL ES?

- OpenGL was just too big for Embedded Systems with limited resources
  - memory footprint, floating point HW

- Create a new, compact API
  - mostly a subset of OpenGL
  - that can still do almost all OpenGL can
  - the work was done at Khronos

- Nokia prototyped such a subset in 2002

OpenGL ES 1.0

- Simplify OpenGL 1.3
  - remove old complex functionality
    - glBegin – glEnd (OUT); vertex arrays (IN)

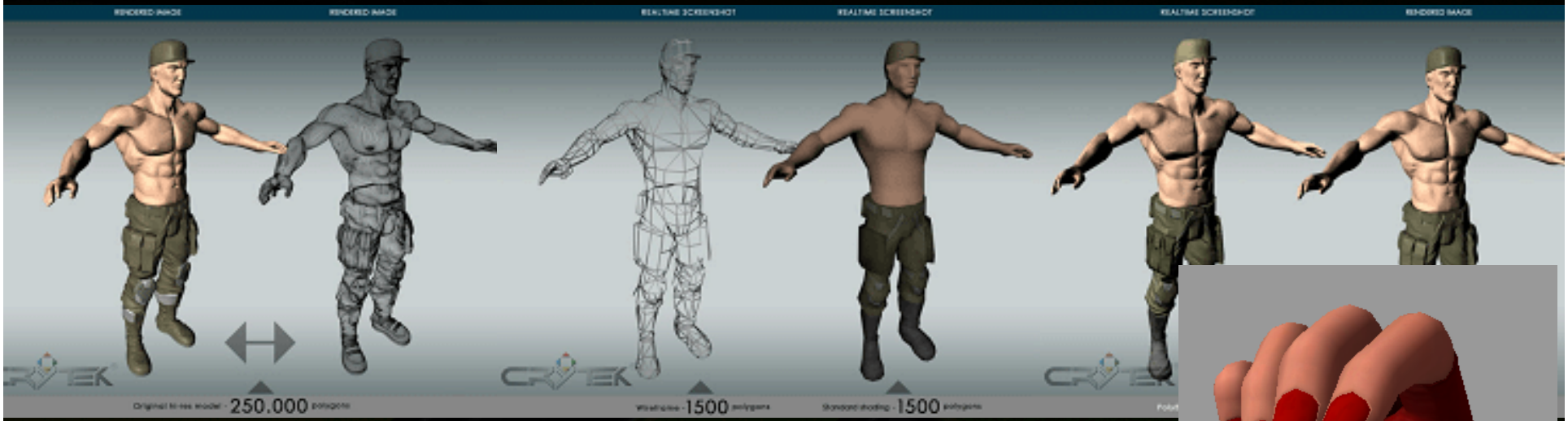
```
glBegin(GL_POLYGON);
glColor3f (1, 0, 0);
glVertex3f(-.5, .5, .5);
glVertex3f( .5, .5, .5);
glColor3f (0, 1, 0);
glVertex3f( .5,-.5, .5);
glVertex3f(-.5,-.5, .5);
glEnd();
```

```
static const GLbyte verts[4 * 3] =
{    -1,  1,  1,      1,  1,  1,
      1, -1,  1,     -1, -1,  1 };
static const GLubyte colors[4 * 3] =
{   255,  0,  0,    255,  0,  0,
      0,255,  0,      0,255,  0 };
glVertexPointer( 3,GL_BYTE,0, verts );
glColorPointerf( 3,GL_UNSIGNED_BYTE,
                 0, colors );
glDrawArrays( GL_TRIANGLE_STRIP,
              0, 4 );
```

  - keep almost all the functionality
    - lights, projections, 2D textures, …
- Shipped in 2004 (mostly SW engines)

OpenGL ES 1.1

- Add features such as bump maps



and skinning (deformable animation)

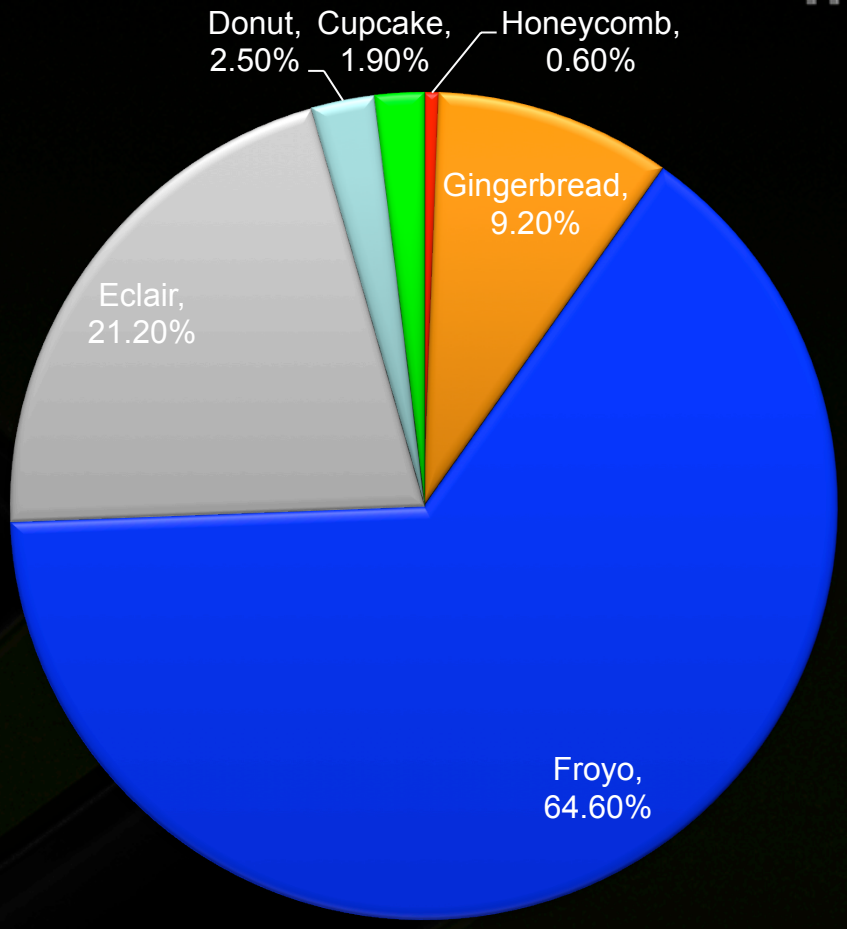
- Shipped in 2005
  - begin of HW accelerated mobile 3D graphics
  - Symbian, iOS, Android, Meego, …

Android SDK

- The main Android SDK is Java
  - the language is very similar to Sun's original Java
    - UI libraries are different from Java SE
    - a different VM (called Dalvik)
- http://developer.android.com/
  - one-stop-shop for Android tools and documentation
- The easiest way to install all the needed tools is
  - http://developer.nvidia.com/tegra-android-development-pack
  - JDK, SDK, NDK, Eclipse, …
  - plus Tegra tools you can ignore if you don't have a Tegra

Many Android versions (1 June 2011)

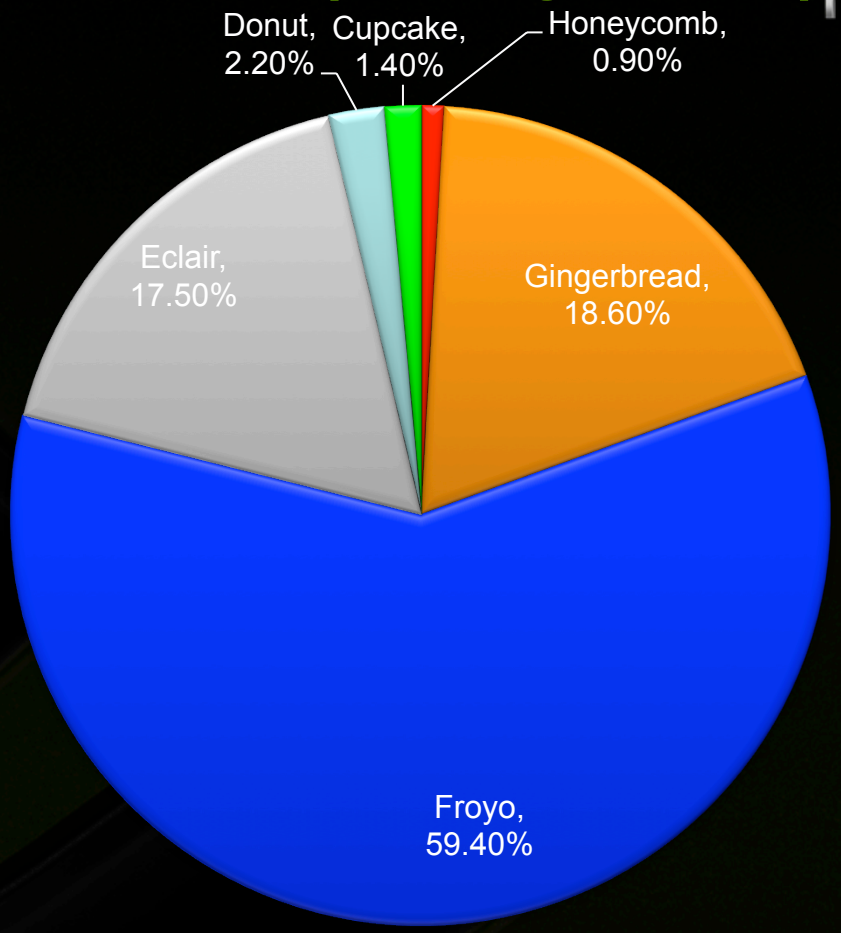
Android Version	Market Share
Honeycomb (3.0, 3.1)	0.6%
Gingerbread (2.3 - 2.3.3)	9.2%
Froyo (2.2)	64.6%
Eclair (2.0/2.1)	21.2%
Donut (1.6)	2.5%
Cupcake (1.5)	1.9%



^ "Platform Versions". *Android Developers*. http://developer.android.com/resources/dashboard/platform-versions.html. "based on the number of Android devices that have accessed Android Market within a 14-day period ending on the data collection date noted below"

Quickly evolving platform (5 July 2011)

Android Version	Market Share
Honeycomb (3.0, 3.1)	0.9%
Gingerbread (2.3 - 2.3.4)	18.6%
Froyo (2.2)	59.4%
Eclair (2.0/2.1)	17.5%
Donut (1.6)	2.2%
Cupcake (1.5)	1.4%

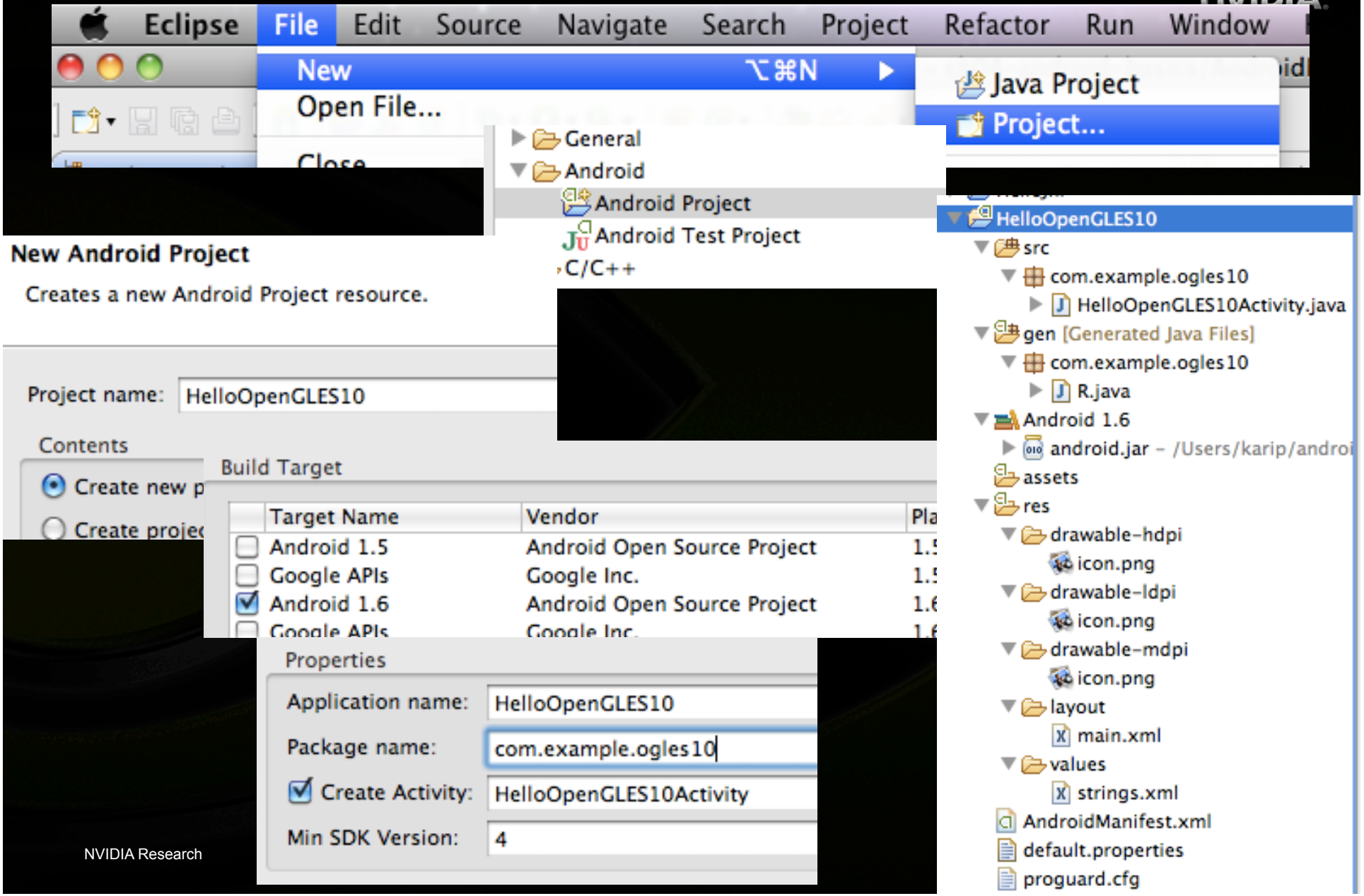


^ "Platform Versions". *Android Developers*. http://developer.android.com/resources/dashboard/platform-versions.html. "based on the number of Android devices that have accessed Android Market within a 14-day period ending on the data collection date noted below"

Hello OpenGL ES 1.0 on Android SDK

- The following OpenGL ES tutorial is available at
  - http://developer.android.com/resources/tutorials/opengl/opengl-es10.html

Create a project (on Eclipse)



Default *.java file

… modify it to …

*HelloOpenGLES10Activity.java

```
package com.example.ogles10;

import android.app.Activity;
import android.content.Context;
import android.opengl.GLSurfaceView;
import android.os.Bundle;

public class HelloOpenGLES10Activity extends Activity {

    private GLSurfaceView mGLView;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Create a GLSurfaceView instance and set it as the ContentView for this Activity.
        mGLView = new HelloOpenGLES10SurfaceView(this);
        setContentView(mGLView);
    }

    @Override
    protected void onPause() {
        super.onPause();
        // The following call pauses the rendering thread. If your OpenGL application is memory intensive,
        // you should consider de-allocating objects that consume significant memory here.
        mGLView.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
        // The following call resumes a paused rendering thread.
        // If you de-allocated graphic objects for onPause() this is a good place to re-allocate them.
        mGLView.onResume();
    }
}

class HelloOpenGLES10SurfaceView extends GLSurfaceView {

    public HelloOpenGLES10SurfaceView(Context context) {
        super(context);
        // Set the Renderer for drawing on the GLSurfaceView
        setRenderer(new HelloOpenGLES10Renderer());
    }
}
```

A couple of new headers

Create a GLSurfaceView and set it as the main content view

Handle Android framework application state changes

Create a renderer for your surface view (in another *.java file)

Create the file / class

HelloOpenGLES10Activity.java | HelloOpenGLES10Renderer.java

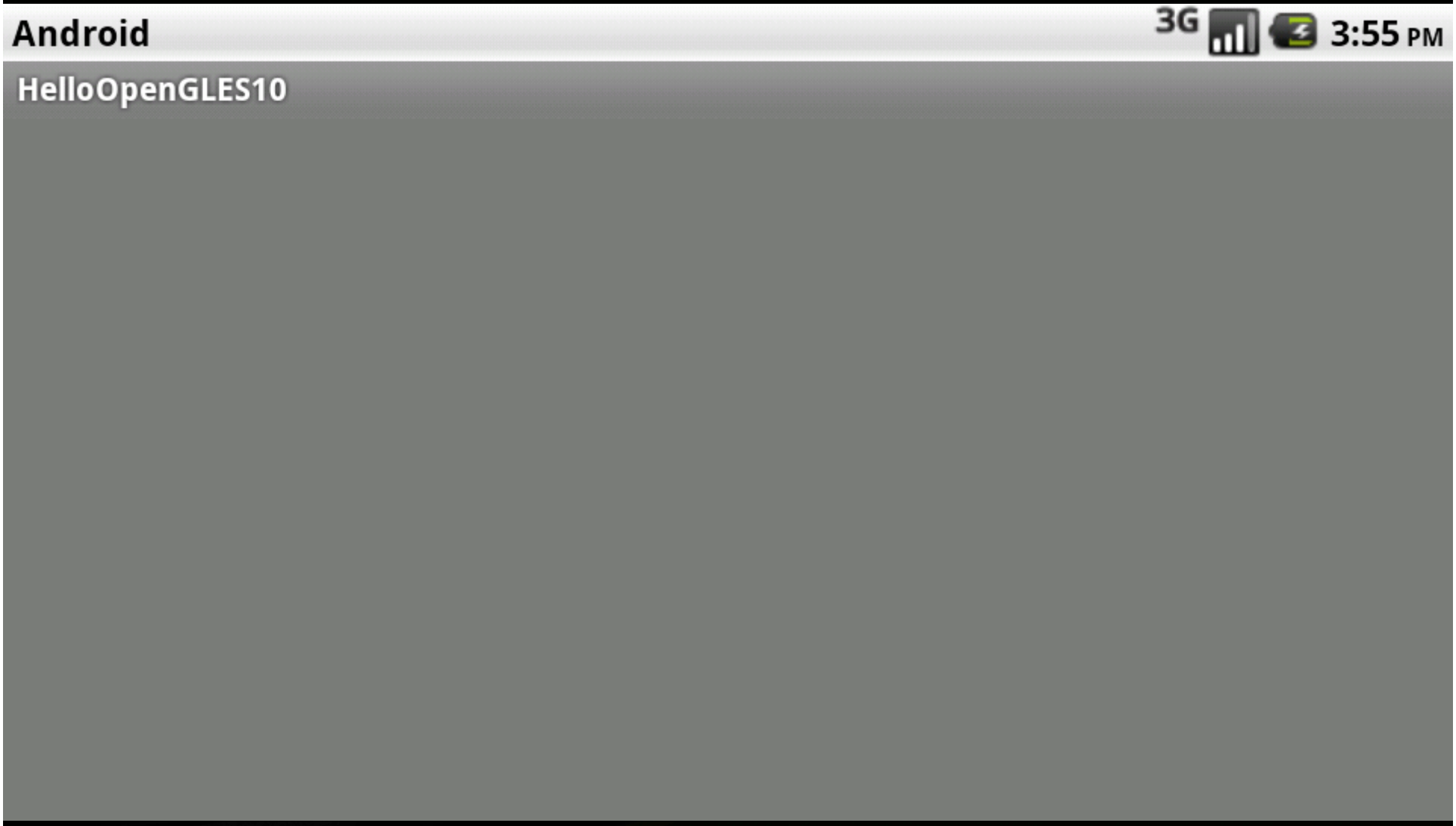
```
package com.example.ogles10;

import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;

import android.opengl.GLSurfaceView;

public class HelloOpenGLES10Renderer implements GLSurfaceView.Renderer {

    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        // Set the background frame color
        gl.glClearColor(0.5f, 0.5f, 0.5f, 1.0f);
    }

    public void onDrawFrame(GL10 gl) {
        // Redraw background color
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT | GL10.GL_DEPTH_BUFFER_BIT);
    }

    public void onSurfaceChanged(GL10 gl, int width, int height) {
        gl.glViewport(0, 0, width, height);
    }

}
```

Called once, to set things up

Called for each redraw event

Called whenever surface changes (e.g., rotate device from portrait to landscape)

Now we can run the app in emulator!



Let's draw a triangle: initialization


```
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.FloatBuffer;
```

```
private FloatBuffer triangleVB;

public void onSurfaceCreated(GL10 gl, EGLConfig config) {
    // Set the background frame color
    gl.glClearColor(0.5f, 0.5f, 0.5f, 1.0f);

    // Define a triangle, by default screen center is [0,0,0],
    // and extends from [-1,-1,0] (lower left) to [1,1,0] (upper right)
    float triangleCoords[] = {
            // X, Y, Z
            -0.5f, -0.25f, 0,
             0.5f, -0.25f, 0,
             0.0f,  0.559016994f, 0
        };

    // initialize vertex Buffer for triangle (# of coordinate values * 4 bytes per float)
    ByteBuffer vbb = ByteBuffer.allocateDirect(triangleCoords.length * 4);
    vbb.order(ByteOrder.nativeOrder()); // use the device hardware's native byte order
    triangleVB = vbb.asFloatBuffer();   // create a floating point buffer from the ByteBuffer
    triangleVB.put(triangleCoords);     // add the coordinates to the FloatBuffer
    triangleVB.position(0);             // set the buffer to read the first coordinate

    // Enable use of vertex arrays
    gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
}
```

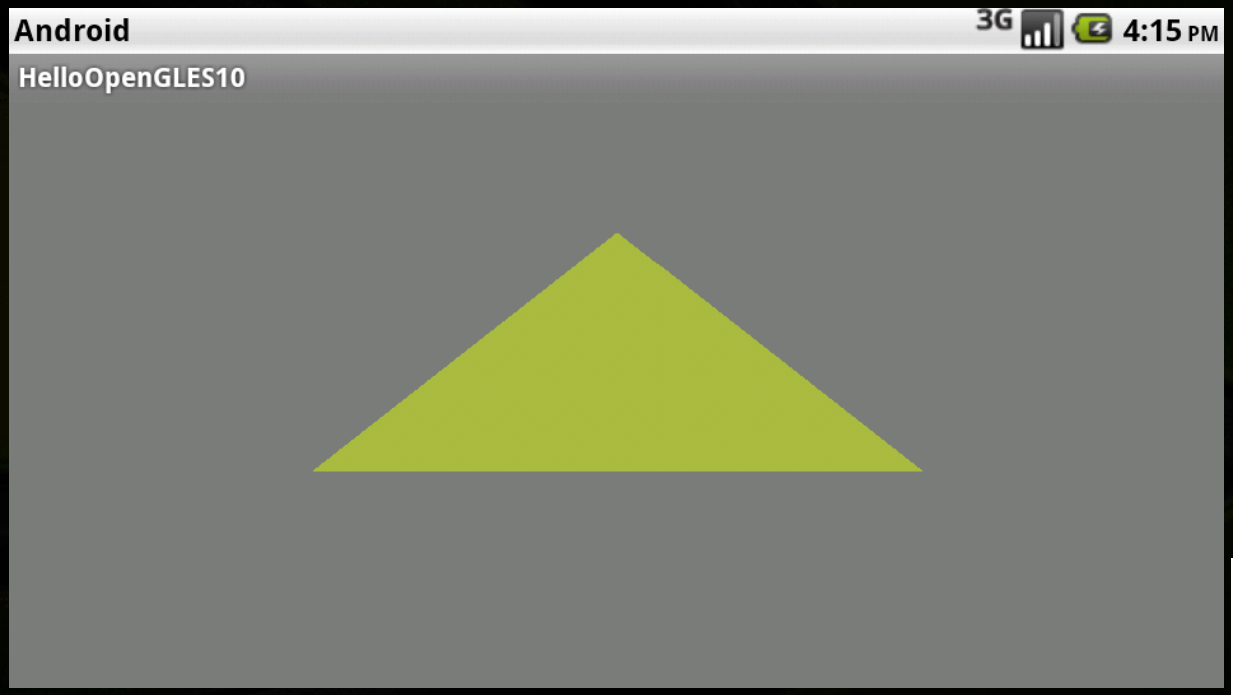
Set background clear color

Define coordinates for an equilateral triangle

Pass the data to a FloatBuffer for passing to GPU

(Re)Draw

```
public void onDrawFrame(GL10 gl) {
    // Redraw background color
    gl.glClear(GL10.GL_COLOR_BUFFER_BIT | GL10.GL_DEPTH_BUFFER_BIT);
    // Draw the triangle
    gl.glColor4f(0.63671875f, 0.76953125f, 0.22265625f, 0.0f);
    gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangleVB);
    gl.glDrawArrays(GL10.GL_TRIANGLES, 0, 3);
}
```

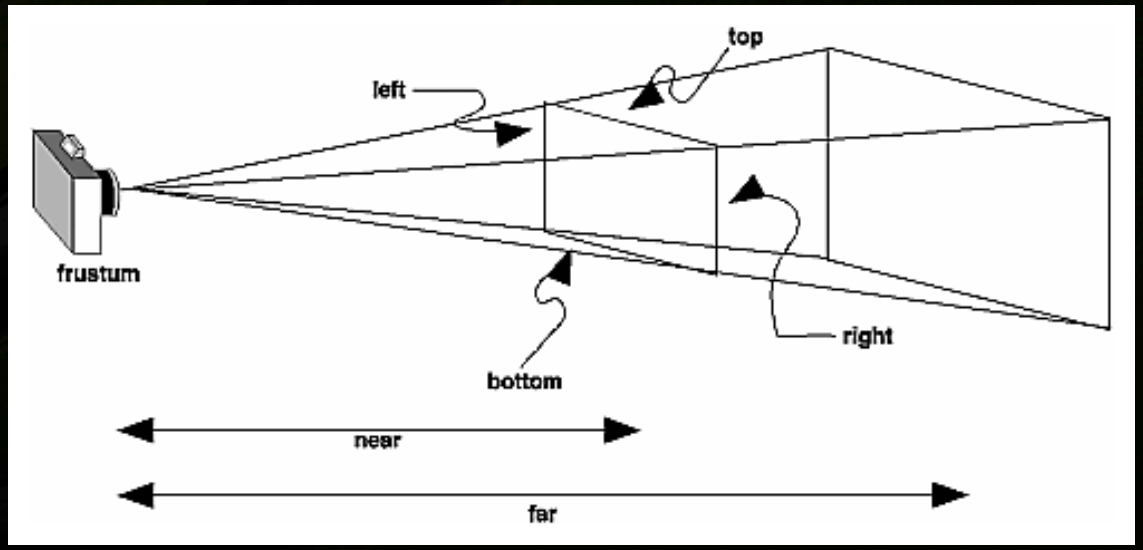



```
// Define a triangle, by default screen center is [0,0,0],
// and extends from [-1,-1,0] (lower left) to [1,1,0] (upper right)
```

```
float triangleCoords[] = {
        // X, Y, Z
        -0.5f, -0.25f, 0,
         0.5f, -0.25f, 0,
         0.0f,  0.559016994f, 0
    };
```

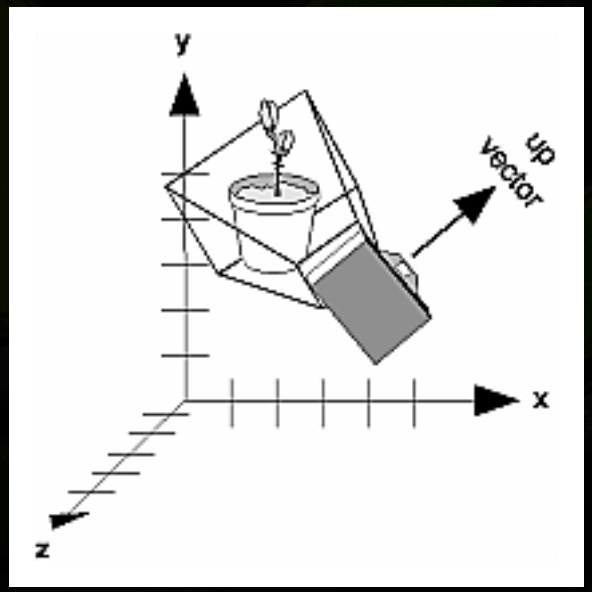
Projection matrix for camera type

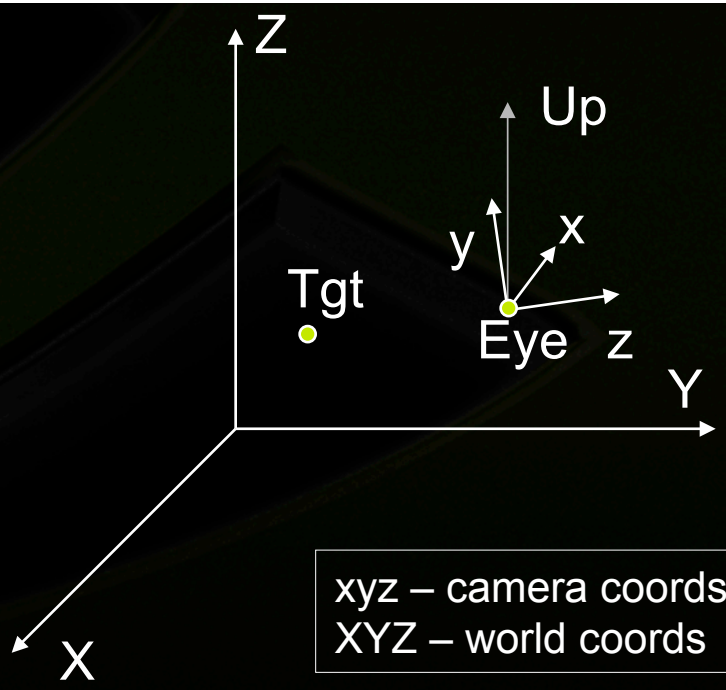
```
public void onSurfaceChanged(GL10 gl, int width, int height) {
    gl.glViewport(0, 0, width, height);
    // make adjustments for screen ratio
    float ratio = (float) width / height;
    gl.glMatrixMode(GL10.GL_PROJECTION);        // set matrix to projection mode
    gl.glLoadIdentity();                        // reset the matrix to its default state
    gl.glFrustumf(-ratio, ratio, -1, 1, 3, 7);  // apply the projection matrix
}
```



Modelview matrix for placing things

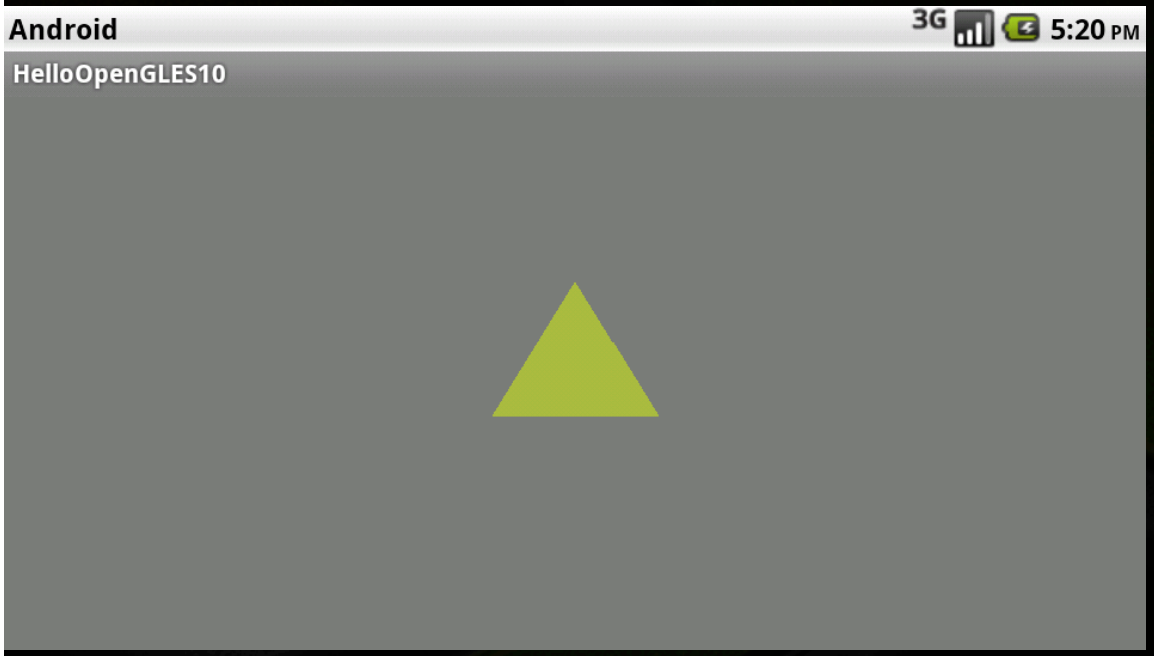
```
public void onDrawFrame(GL10 gl) {
    // Redraw background color
    gl.glClear(GL10.GL_COLOR_BUFFER_BIT | GL10.GL_DEPTH_BUFFER_BIT);

    // Set GL_MODELVIEW transformation mode
    gl.glMatrixMode(GL10.GL_MODELVIEW);
    gl.glLoadIdentity();   // reset the matrix to its default state

    // When using GL_MODELVIEW, you must set the view point
    //          EyeX, EyeY, EyeZ,  TgtX, TgtY, TgtZ,  UpX, UpY, UpZ
    GLU.gluLookAt(gl,  0, 0, -5,  0f, 0f, 0f,        0f, 1.0f, 0.0f);
```





xyz – camera coords
XYZ – world coords

No more stretching and squeezing based on orientation change

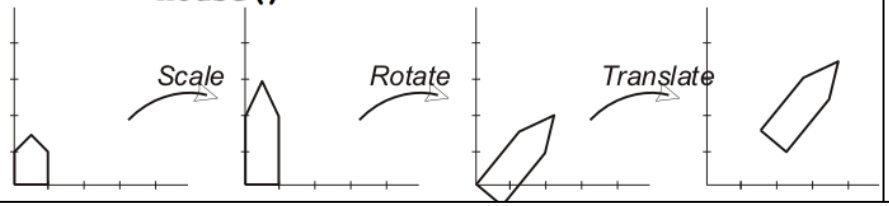


On OpenGL transformations

Instances and transformations

- An instance of a house is transformed by an instance transformation

```
glMatrixMode( GL_MODELVIEW )
glLoadIdentity()
glTranslatef( ... )
glRotatef( ... )
glScalef( ... )
house()
```

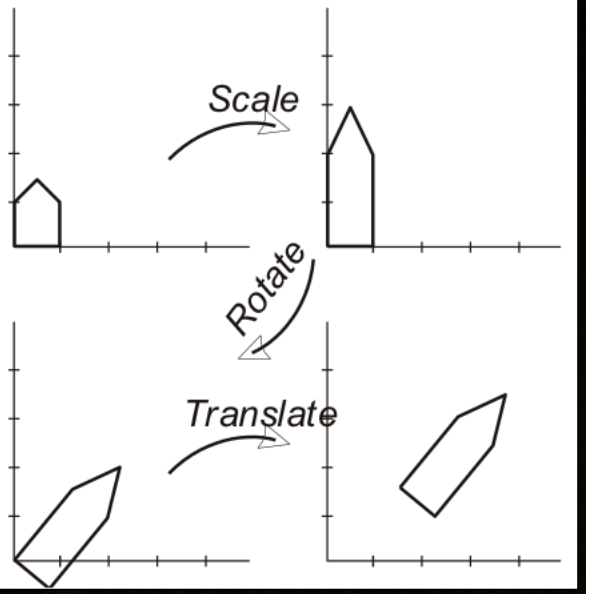


Global, fixed coordinate system

- OpenGL's transforms, logical as they may be, still *seem backwards*

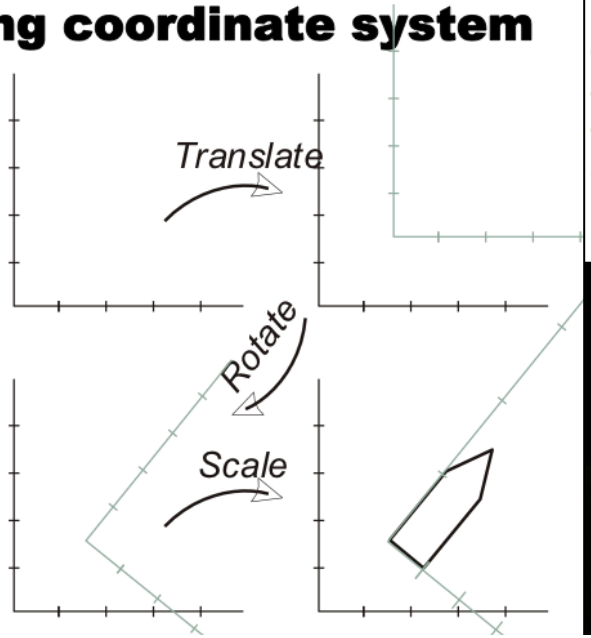
```
glLoadIdentity()
glTranslatef( ... )
glRotatef( ... )
glScalef( ... )
house()
```

They are, if you think of them as transforming the object in a **fixed** coordinate system

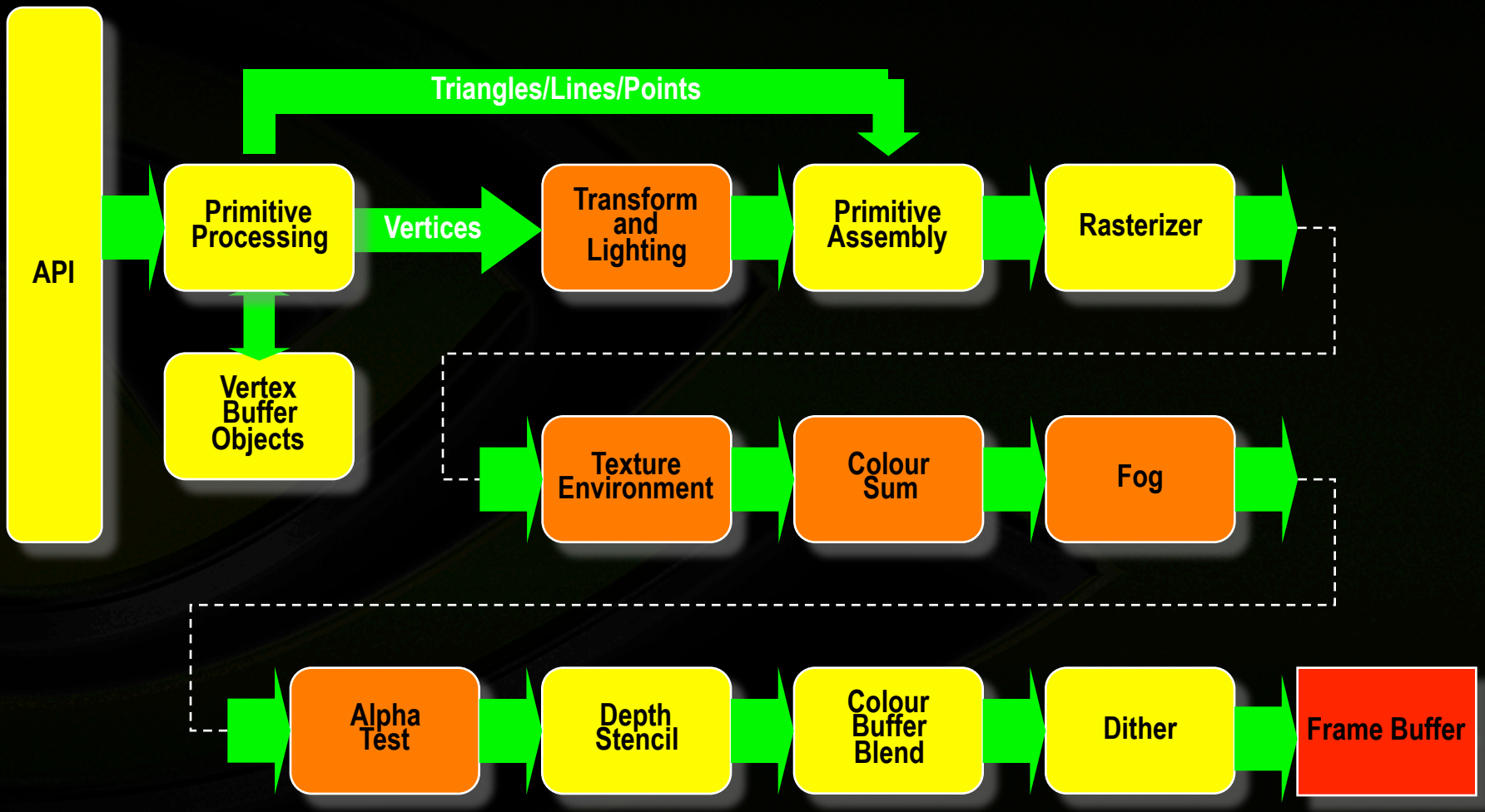


Local, changing coordinate system

- Another way to view transformations is as affecting a *local coordinate system* that the primitive is drawn in
- Now the transforms appear in the "right" order

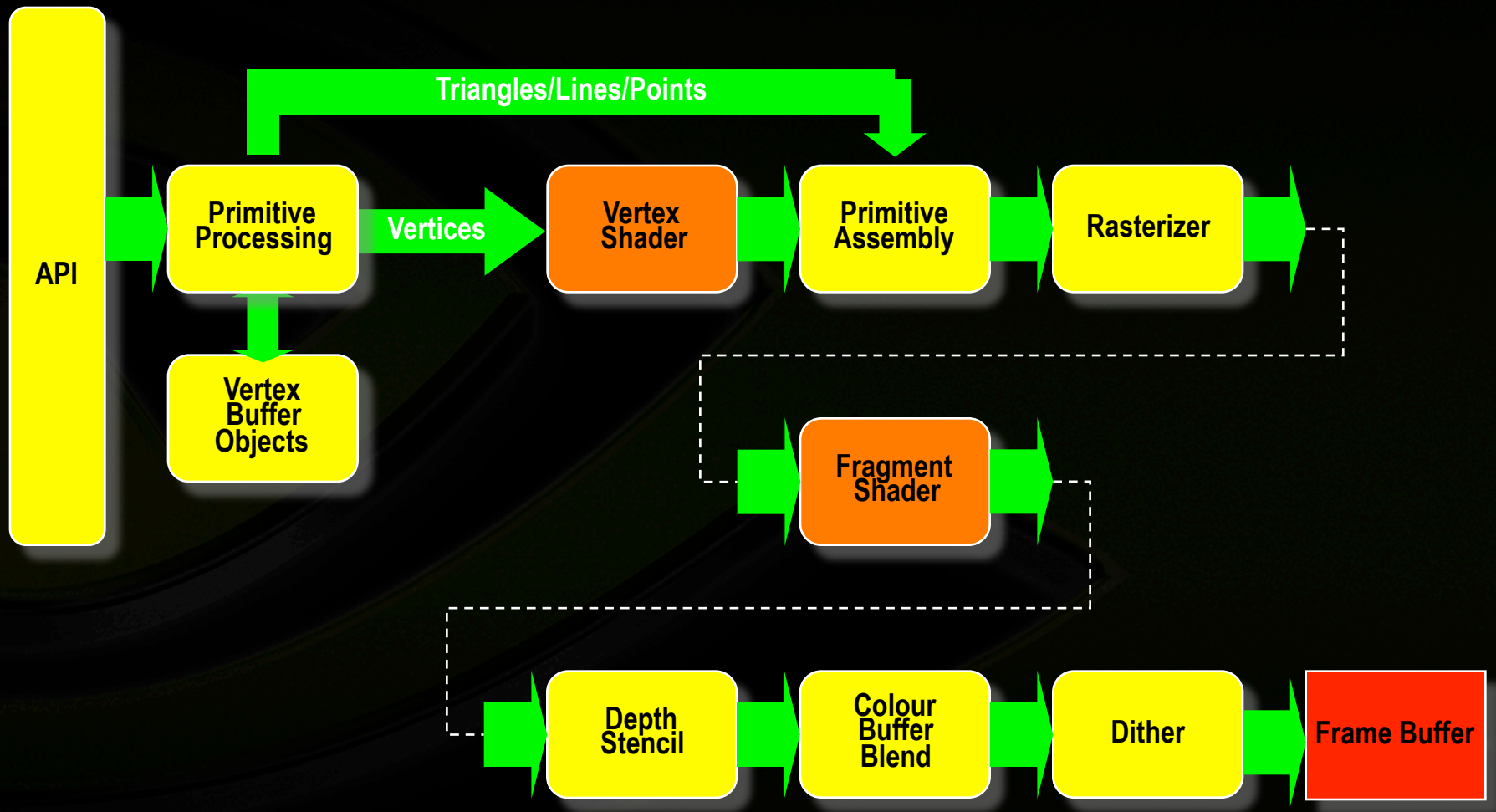


OpenGL ES 1.x Fixed Function pipe



Fixed-function graphics in games

OpenGL ES 2.0 Programmable pipe
(introduced in 2007)



Shaders in games

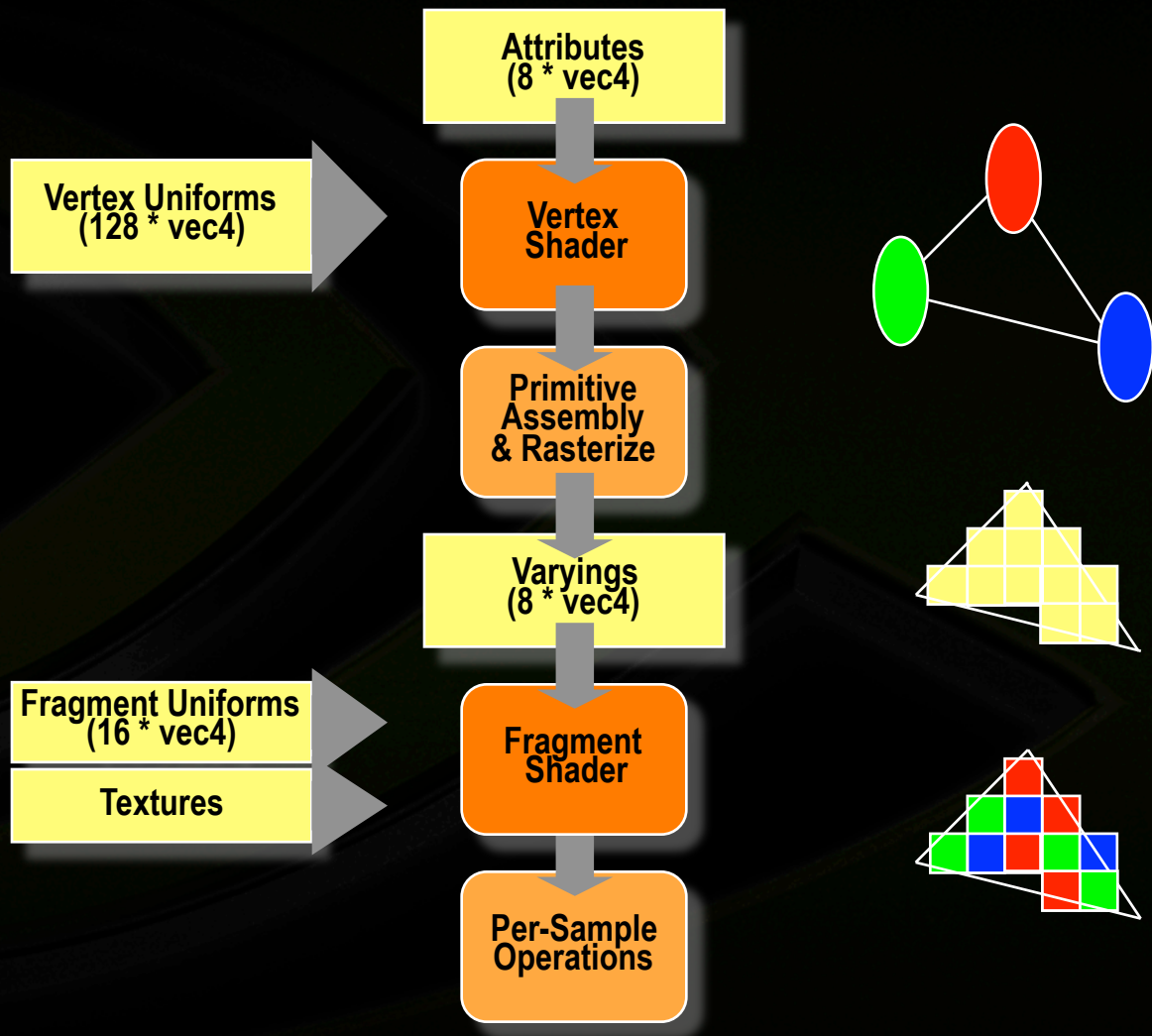
The difference?

- Fixed-function graphics
  - simple illumination models
  - evaluate at vertices, interpolate in between
- Shaders
  - vertex shader
    - smooth deformations and morphing
  - fragment shader
    - per-pixel illumination effects
    - arbitrary illumination models

Not backwards compatible

- ES 2.0 is a very bare-bones API
- Setup
- Input of per-object constants (uniforms)
  - no matrix calculation support in the API
    - do it on CPU with other utility APIs
- Input of per-vertex data (attributes)
  - no special property types
    - normals, texture coordinates, …
  - it's up to the shaders to interpret what the numbers mean
- And the shaders of course
  - sources as strings, compiled and linked on the fly
  - connect CPU variables with shader variables

Programmer's model



The Vertex Shader can do:

- Modify any of the attributes of a vertex
- Examples (old stuff, ES 1.1 could do):
  - Transformation of position and normals
    - using model-view and projection matrices
  - Texture coordinate generation and transformation
  - Per-vertex lighting
- Examples (new stuff):
  - Arbitrary vertex morphing and skinning
  - Advanced lighting
    - per-vertex more complex lighting models
    - calculation of values used later for per-pixel lighting
  - Other per vertex data (density for physics, …)

The Vertex Shader cannot do:

- Anything that
  - requires information from more than one vertex
  - for example
    - any triangle operations (e.g. clipping, culling)
    - tessellation of meshes to denser meshes
  - accesses color buffer

- That is: “One comes in, one goes out”

The Fragment Shader can do:

- Calculate new colors per pixel
  - texture blending
  - fog
- Dependent textures
- Pixel discard
  - alpha testing
- Bump and environment mapping
- Render to multiple render targets (MTR) at once
  - think of running the vertex part only once, but for the same geometry, producing different intermediate results to different textures at once

The Fragment Shader cannot do:

- Read from any buffer
  - including the previous color
- Write to any buffer
  - except provide the new color
  - which is blended by the later stages of fixed-funtion pipe
- Why the limitations?
  Now the implementation can
  - rasterize different elements in parallel
  - hide the latency of frame buffer access for blending

Android and Native Code

- Native code adds some complexity
  - but unlocks peak performance (games, image processing)
  - makes porting existing (C/C++) code easier
  - reduces portability
- A set of cross-compiler toolchains
  - gcc for ARM
  - hosted on Windows, Linux, or Mac
- JNI, the Java/Native Interface
  - connecting native code to Java
- Gingerbread (Android 2.3) introduced NativeActivity
  - avoids the need for any Java code

OpenGL ES 2.0 on Android NDK

- Android 2.2 (Froyo) introduced ES 2.0 on Java
  - but still not on emulators, only on device
- We show a C example using NDK
- The following example comes from
  - http://developer.nvidia.com/tegra-android-development-pack
  - using its app_create.sh script

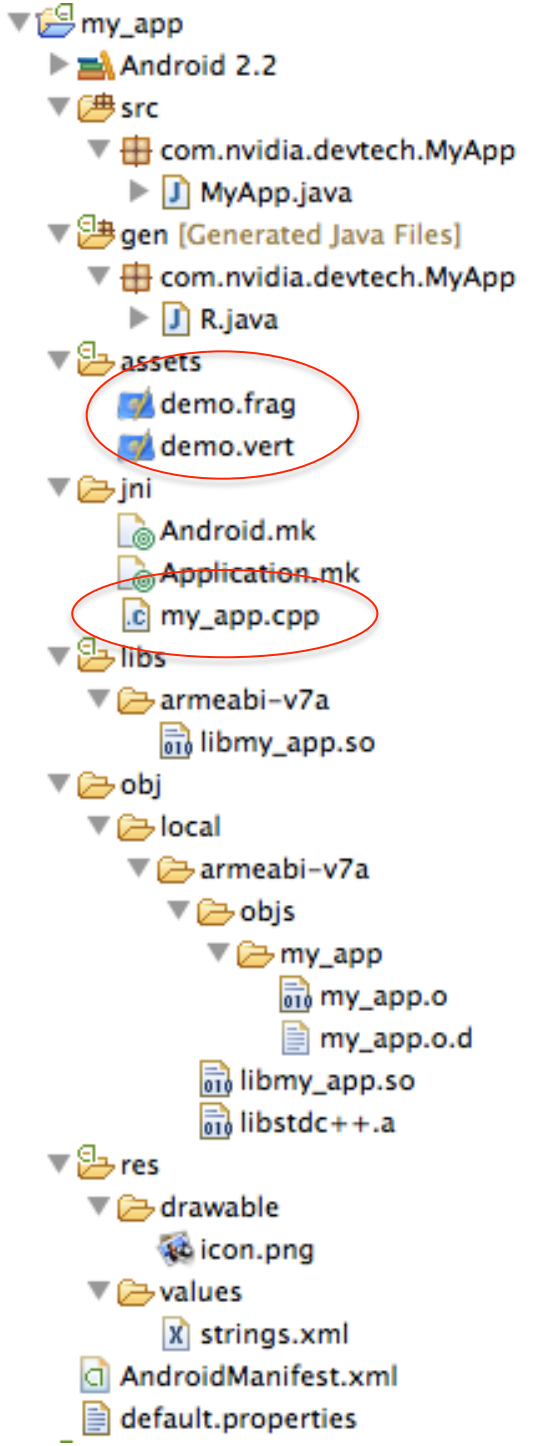
Tegra pack examples

- Import all sample projects to Eclipse workspace
  - set up environment variables etc. following instructions on the pdf file
  - build all, try on a device
    here's es2_globe

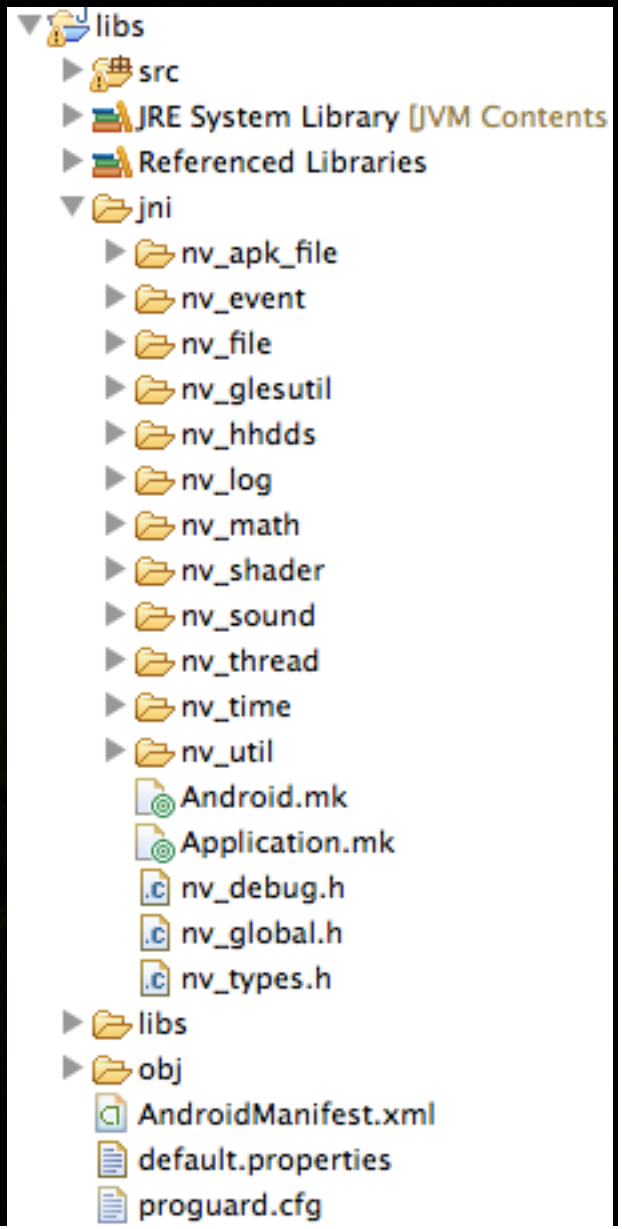
- Create your own sample app

```
karip@mac:~/NVPACK/TDK_Samples/Android_NVIDIA_samples_2_20110315/tools/app_create$
./app_create.sh my_app MyApp basic
INFO: Java/native app being placed in apps
creating destination directory tree ../../apps/my_app
copying files to destination tree
```

- Build and try it



- The project
- Also need a library project for various utilities



Java side: extend NvGLES2Activity

MyApp.java

```
// File:            tools\app_create\basic\template.java

package com.nvidia.devtech.MyApp;

import com.nvidia.devtech.*;
import android.view.MotionEvent;
import android.view.KeyEvent;

public class MyApp extends NvGLES2Activity
{
    @Override
    public native boolean init();

    @Override
    public native boolean render(float drawTime, int drawWidth, int drawHeight, boolean forceRedraw);

    @Override
    public native void cleanup();

    @Override
    public native boolean inputEvent(int action, float x, float y, MotionEvent e);

    @Override
    public native boolean keyEvent(int action, int keyCode, KeyEvent e);

    static
    {
        System.loadLibrary("my_app");
    }
}
```

Matching part from C side

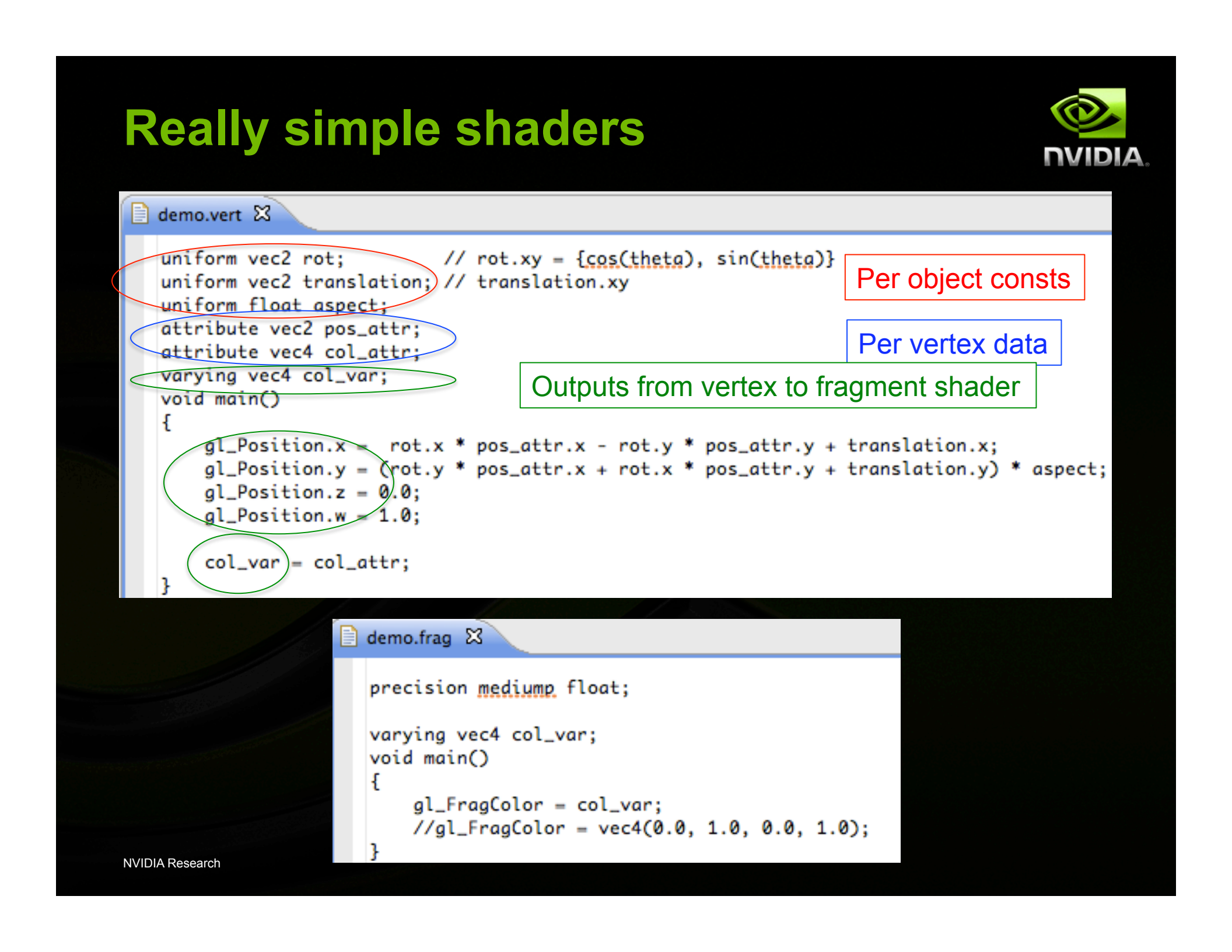

```
jint JNI_OnLoad(JavaVM* vm, void* reserved)
{
    JNIEnv *env;

    NVThreadInit(vm);

    if (vm->GetEnv((void**) &env, JNI_VERSION_1_4) != JNI_OK) {
        DEBUG("Failed to get the environment using GetEnv()");
        return -1;
    }
    JNINativeMethod methods[] = {
        { "init",       "()Z",                                  (void *) init },
        { "render",     "(FIIZ)Z",                              (void *) render },
        { "inputEvent", "(IFFLandroid/view/MotionEvent;)Z", (void *) inputEvent },
        { "keyEvent",   "(IILandroid/view/KeyEvent;)Z",     (void *) keyEvent },
        { "cleanup",    "()V",                                  (void *) cleanup }
    };
    jclass k = (env)->FindClass ("com/nvidia/devtech/MyApp/MyApp");
    (env)->RegisterNatives(k, methods, 5);

    NVTimeInit();

    return JNI_VERSION_1_4;
}
```

Match the C functions, with their types, to Java functions

Initialize shaders and attributes


```
#define ROOT_3_OVER_2 0.8660254f
#define ROOT_3_OVER_6 (ROOT_3_OVER_2/3.0f)
static GLfloat s_vert[6] = { 0.5f, -ROOT_3_OVER_6,
                            -0.5f, -ROOT_3_OVER_6,
                             0.0f,  ROOT_3_OVER_2 - ROOT_3_OVER_6 };
static GLfloat s_col [12] = { 1.0, 0.0, 0.0, 1.0,
                              0.0, 1.0, 0.0, 1.0,
                              0.0, 0.0, 1.0, 1.0};

jboolean init(JNIEnv*  env, jobject  thiz)
{
    nv_shader_init();
    DEBUG("GL_VENDOR: %s", glGetString(GL_VENDOR));
    DEBUG("GL_VERSION: %s", glGetString(GL_VERSION));
    DEBUG("GL_RENDERER: %s", glGetString(GL_RENDERER));

    s_prog = nv_load_program("demo");

    s_angle = 0;
    s_translationX = 0;
    s_translationY = 0;

    return JNI_TRUE;
}
```

Really simple shaders



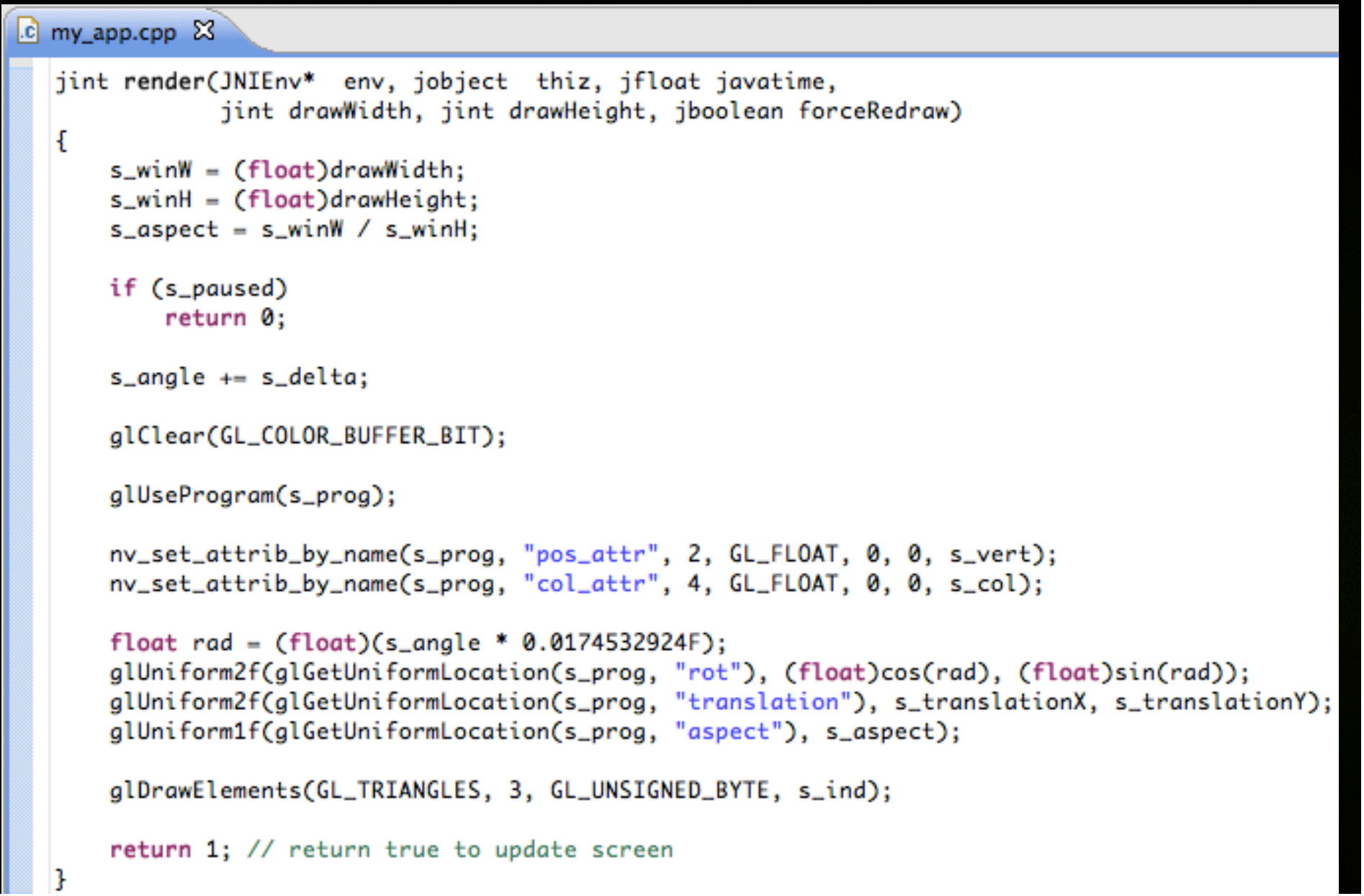
demo.vert

```
uniform vec2 rot;          // rot.xy = {cos(theta), sin(theta)}
uniform vec2 translation; // translation.xy
uniform float aspect;
attribute vec2 pos_attr;
attribute vec4 col_attr;
varying vec4 col_var;
void main()
{
    gl_Position.x =  rot.x * pos_attr.x - rot.y * pos_attr.y + translation.x;
    gl_Position.y = (rot.y * pos_attr.x + rot.x * pos_attr.y + translation.y) * aspect;
    gl_Position.z = 0.0;
    gl_Position.w = 1.0;

    col_var = col_attr;
}
```


demo.frag

```
precision mediump float;

varying vec4 col_var;
void main()
{
    gl_FragColor = col_var;
    //gl_FragColor = vec4(0.0, 1.0, 0.0, 1.0);
}
```

Rendering callback function



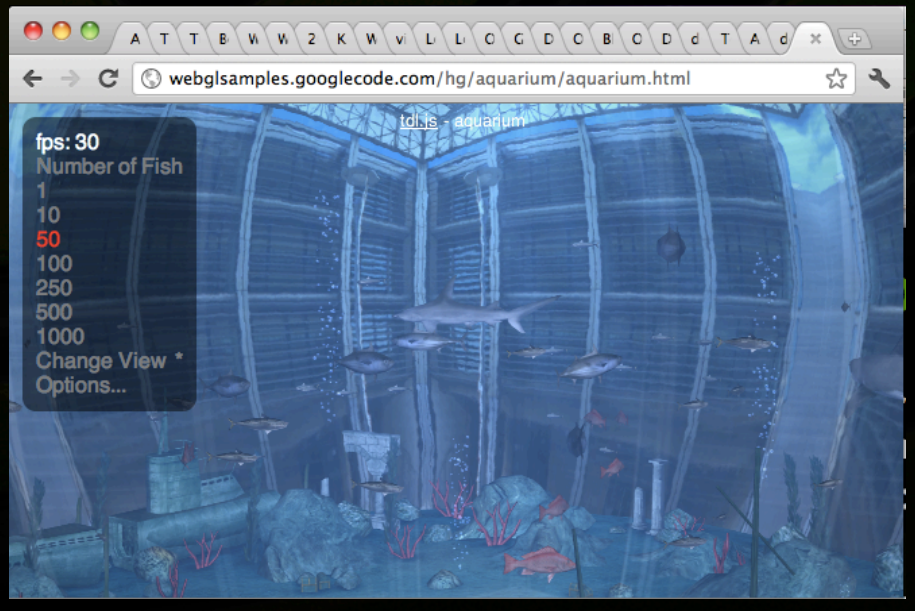
Cleanup, and touch input processing

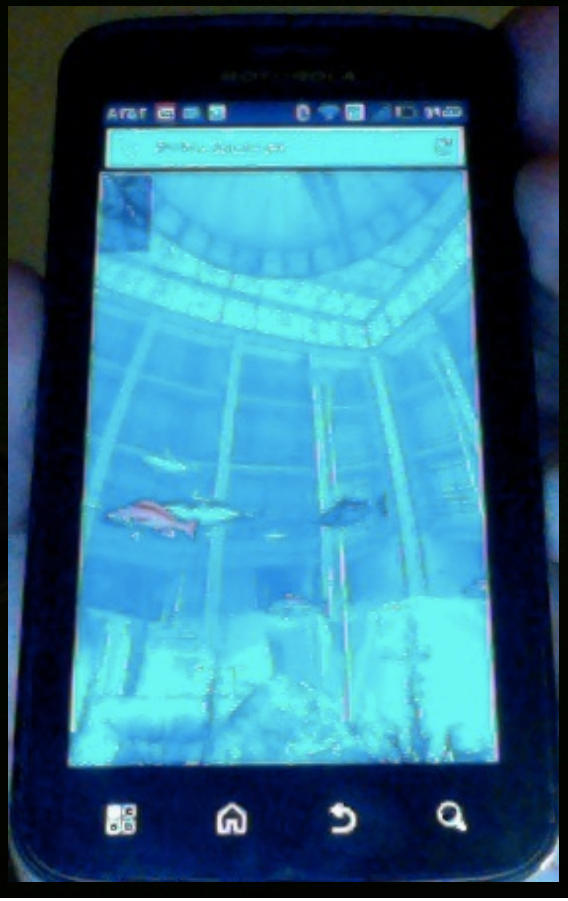
my_app.cpp

```
void cleanup(JNIEnv*  env)
{
    DEBUG("cleanup!!!");

    if(s_prog)
    {
        glDeleteProgram(s_prog);
        s_prog = 0;
    }
}

jboolean inputEvent(JNIEnv*  env, jobject  thiz, jint action, jfloat mx, jfloat my)
{
    DEBUG("inputEvent!!!");
    s_translationX = mx * 2.0f / s_winW - 1.0f;
    s_translationY = (1.0f - my * 2.0f / s_winH) / s_aspect;

    return JNI_TRUE;
}
```

WebGL: OpenGL ES 2.0 in JavaScript

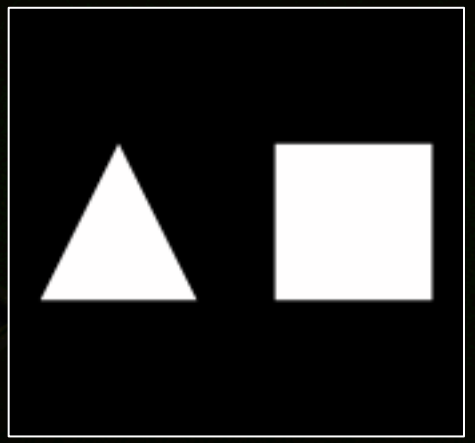
- The only truly cross-platform way to create applications is to use web technologies
- WebGL provides OpenGL ES 2.0 in a browser
  - Supported by many desktop browsers
    - Chrome, Firefox, Safari, Opera
  - Coming to also on smartphones
    - Firefox on Android runs WebGL





WebGL tutorial: http://learningwebgl.com


```
<html>  <head>
<title>Learning WebGL &mdash; lesson 1</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
…   // shaders, mini matrix library
<script type="text/javascript">
…   // more functions

    function webGLStart() {
        var canvas = document.getElementById("lesson01-canvas");
        initGL(canvas);
        initShaders();
        initBuffers();
        gl.clearColor(0.0, 0.0, 0.0, 1.0);
        gl.enable(gl.DEPTH_TEST);
        drawScene();
    }
</script>
</head>

<body onload="webGLStart();">
    <canvas id="lesson01-canvas" style="border: none;" width="500" height="500">
    </canvas>
</body> </html>
```



Initialize gl context


```
var gl;

function initGL(canvas) {
    try {
        gl = canvas.getContext("experimental-webgl");
        gl.w = canvas.width;
        gl.h = canvas.height;
    } catch (e) {
    }
    if (!gl) {
        alert("Could not initialise WebGL, sorry :-(");
    }
}
```

Shaders: vertex, fragment


```
<script id="shader-vs" type="x-shader/x-vertex">
    attribute vec3 aVertexPosition;
    uniform mat4 uMVMatrix;
    uniform mat4 uPMatrix;

    void main(void) {
        gl_Position = uPMatrix * uMVMatrix * vec4(aVertexPosition, 1.0);
    }
</script>

<script id="shader-fs" type="x-shader/x-fragment">
    #ifdef GL_ES
    precision highp float;
    #endif

    void main(void) {
        gl_FragColor = vec4(1.0, 1.0, 1.0, 1.0);
    }
</script>
```

Compile a shader


```
function getShader(gl, id) {
    var shaderScript = document.getElementById(id);
    if (!shaderScript) { return null; }

    var shader;
    if (shaderScript.type == "x-shader/x-fragment") {
        shader = gl.createShader(gl.FRAGMENT_SHADER);
    } else if (shaderScript.type == "x-shader/x-vertex") {
        shader = gl.createShader(gl.VERTEX_SHADER);
    } else { return null; }

    var range = document.createRange();
    range.selectNodeContents(shaderScript);
    gl.shaderSource(shader, range.toString());
    range.detach();
    gl.compileShader(shader);

    if (!gl.getShaderParameter(shader, gl.COMPILE_STATUS)) {
        alert(gl.getShaderInfoLog(shader));
        return null;
    }
    return shader;
}
```

Compile and link GL program; get attribute & uniform locations


```
var prg; // shader program

function initShaders() {
    prg = gl.createProgram();
    gl.attachShader(prg, getShader(gl, "shader-vs"));
    gl.attachShader(prg, getShader(gl, "shader-fs"));
    gl.linkProgram(prg);

    if (!gl.getProgramParameter(prg, gl.LINK_STATUS)) {
        alert("Could not initialise shaders");
    }

    gl.useProgram(prg);

    prg.vtxPos = gl.getAttribLocation(prg, "aVertexPosition");
    gl.enableVertexAttribArray(prg.vtxPos);

    prg.pMatrixUniform  = gl.getUniformLocation(prg, "uPMatrix");
    prg.mvMatrixUniform = gl.getUniformLocation(prg, "uMVMatrix");
}
```

Initialize vertex position buffers


```
var tri; // triangleVertexPositionBuffer
var sqr; // squareVertexPositionBuffer

function initBuffers() {
    tri = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, tri);
    var vertices = [ 0.0,  1.0,  0.0,
                    -1.0, -1.0,  0.0,
                     1.0, -1.0,  0.0 ];
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);
    tri.itemSize = 3;  tri.numItems = 3;

    sqr = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, sqr);
    vertices = [ 1.0,  1.0,  -1.0,
                -1.0,  1.0,  -1.0,
                 1.0, -1.0,  -1.0,
                -1.0, -1.0,  -1.0 ];
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);
    sqr.itemSize = 3;  sqr.numItems = 4;
}
```

Include a simple matrix utility library

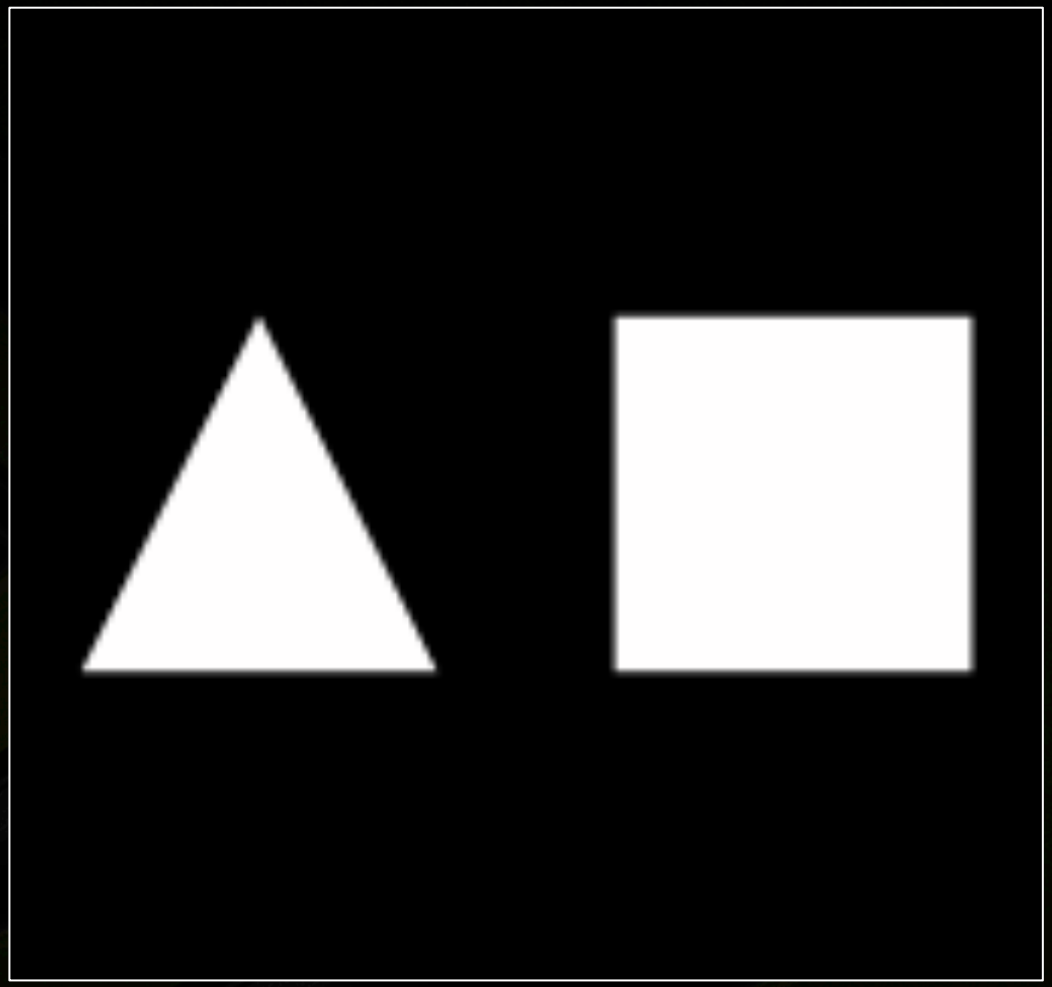
- OpenGL ES 2.0 does not include any matrix utilities
  - idea is that matrix math happens on CPU
  - OpenGL ES 2.0 concentrates on things running on GPU
  - it's easy to write such utility libraries

- Include such a library from a file in the same folder

```
<script type="text/javascript"
    src="glMatrix-0.9.5.min.js">
</script>
```

Finally, draw the scene


```
function drawScene() {
    gl.viewport(0, 0, gl.w, gl.h);
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

    var pMatrix = mat4.create();
    mat4.perspective(45, gl.w / gl.h, 0.1, 100.0, pMatrix);
    gl.uniformMatrix4fv(prg.pMatrixUniform, false, pMatrix);

    var mvMatrix = mat4.create();  mat4.identity(mvMatrix);
    mat4.translate(mvMatrix, [-1.5, 0.0, -7.0]);
    gl.uniformMatrix4fv(prg.mvMatrixUniform, false, mvMatrix);

    gl.bindBuffer(gl.ARRAY_BUFFER, tri);
    gl.vertexAttribPointer(prg.vtxPos, tri.itemSize, gl.FLOAT, false, 0, 0);
    gl.drawArrays(gl.TRIANGLES, 0, tri.numItems);

    mat4.translate(mvMatrix, [3.0, 0.0, 0.0]);
    gl.uniformMatrix4fv(prg.mvMatrixUniform, false, mvMatrix);
    gl.bindBuffer(gl.ARRAY_BUFFER, sqr);
    gl.vertexAttribPointer(prg.vtxPos, sqr.itemSize, gl.FLOAT, false, 0, 0);
    gl.drawArrays(gl.TRIANGLE_STRIP, 0, sqr.numItems);
}
```



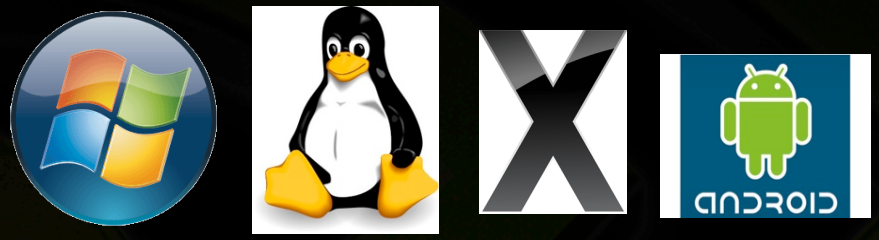
Camera: the “I” of visual I/O

- Graphics is one half of the visual I/O
  - the output
- Camera provides the input
  - capture textures
  - affect the application
- New Tegra board supports cameras
  - Back: Stereo 5 MP
  - Front: 2 MP
  - SW: OpenCV & FCam

OpenCV

Thousands of Developers, Cross Platform API

- Open standard for Computer Vision



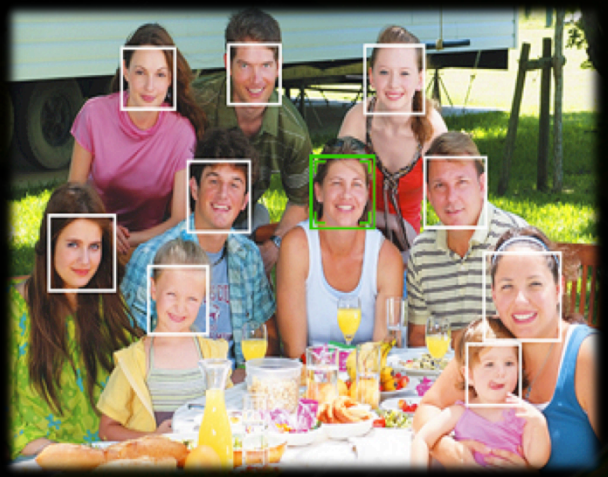
- 12 years old, professionally developed
  - Over 3 Million Downloads!
- > 500 Algorithms

Common API for Server, Workstation, Desktop and now Mobile Platforms!

Computer Vision = Smart Photography

Get the right shot, automatically

Face Detection



Scene Classification

Stabilization

Computer Vision = New Applications

Augmented Reality

Augmented Reality Ghost Hunter (Argh)

Wordlens

Google Goggles

Gesture interfaces

OpenCV functionality overview

General Image Processing

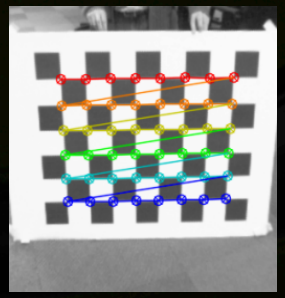
Segmentation

Machine Learning, Detection

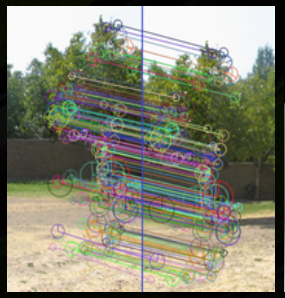
Image Pyramids

Transforms

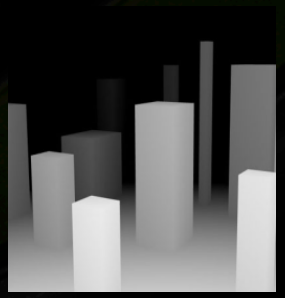
Fitting



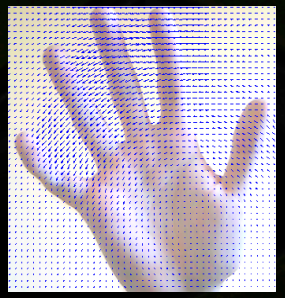
Camera Calibration



Features



Depth Maps



Optical Flow

Inpainting

Tracking

OpenCV for Android

Optimized for ARM, Tegra, & Android

- **Install from**
  - **http://opencv.itseez.com/doc/tutorials/introduction/android_binary_package/android_binary_package.html**

Tutorial: Android Camera

- Part of the Android OpenCV distribution
- Get camera image
- Display it

Tutorial: Add OpenCV

- The second part of the tutorial adds OpenCV functionality
  - real-time Canny edge detector from the input image

```java
@Override
public void surfaceChanged(SurfaceHolder _holder, int format, int width, int height) {
    super.surfaceChanged(_holder, format, width, height);

    synchronized (this) {
        // initialize Mats before usage
        mYuv = new Mat(getFrameHeight() + getFrameHeight() / 2, getFrameWidth(), CvType.CV_8UC1);
        mGraySubmat = mYuv.submat(0, getFrameHeight(), 0, getFrameWidth());

        mRgba = new Mat();
        mIntermediateMat = new Mat();
    }
}

@Override
protected Bitmap processFrame(byte[] data) {
    mYuv.put(0, 0, data);

    switch (Sample1Java.viewMode) {
    case Sample1Java.VIEW_MODE_GRAY:
        Imgproc.cvtColor(mGraySubmat, mRgba, Imgproc.COLOR_GRAY2RGBA, 4);
        break;
    case Sample1Java.VIEW_MODE_RGBA:
        Imgproc.cvtColor(mYuv, mRgba, Imgproc.COLOR_YUV420sp2RGB, 4);
        Core.putText(mRgba, "OpenCV + Android", new Point(10, 100),
                3/* CV_FONT_HERSHEY_COMPLEX */, 2, new Scalar(255, 0, 0, 255), 3);
        break;
    case Sample1Java.VIEW_MODE_CANNY:
        Imgproc.Canny(mGraySubmat, mIntermediateMat, 80, 100);
        Imgproc.cvtColor(mIntermediateMat, mRgba, Imgproc.COLOR_GRAY2BGRA, 4);
        break;
    }

    Bitmap bmp = Bitmap.createBitmap(getFrameWidth(), getFrameHeight(), Bitmap.Config.ARGB_8888);

    if (Utils.matToBitmap(mRgba, bmp))
        return bmp;

    bmp.recycle();
    return null;
}
```

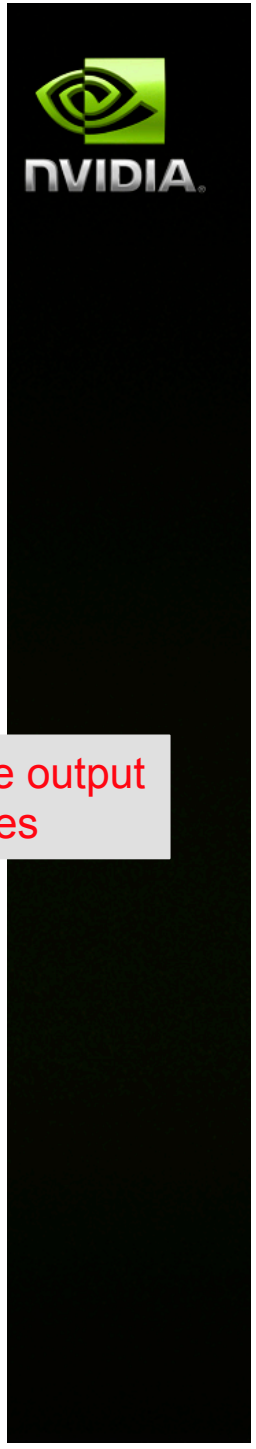
Mat is a 2D storage for Matrices, image data, etc.

Yuv data is organized so that the gray scale image comes first, followed by the colors

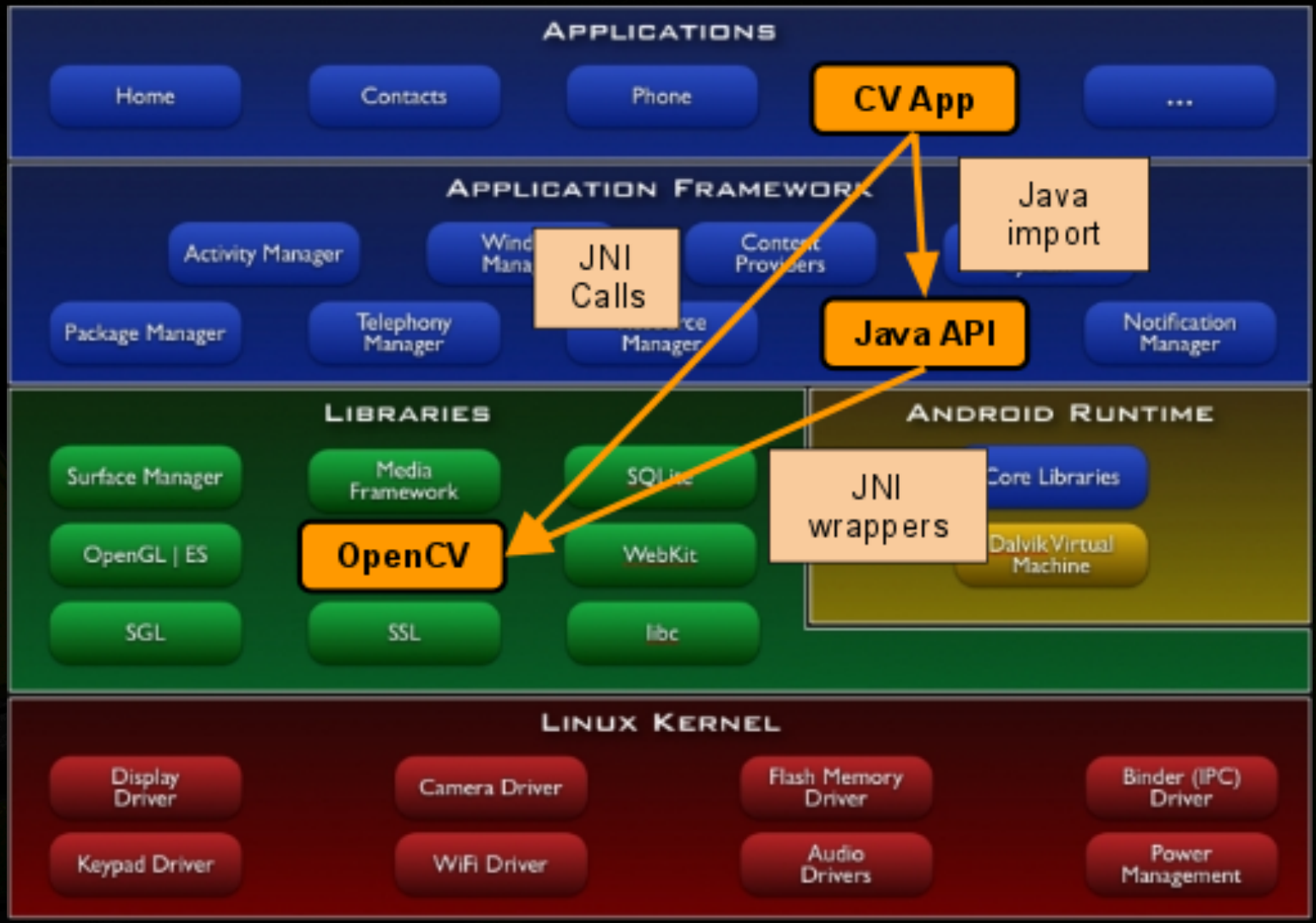
Put input data into mYuv Mat

mRgba Mat is the output of any of the cases

convert to Bitmap for display



OpenCV supports Java and Native

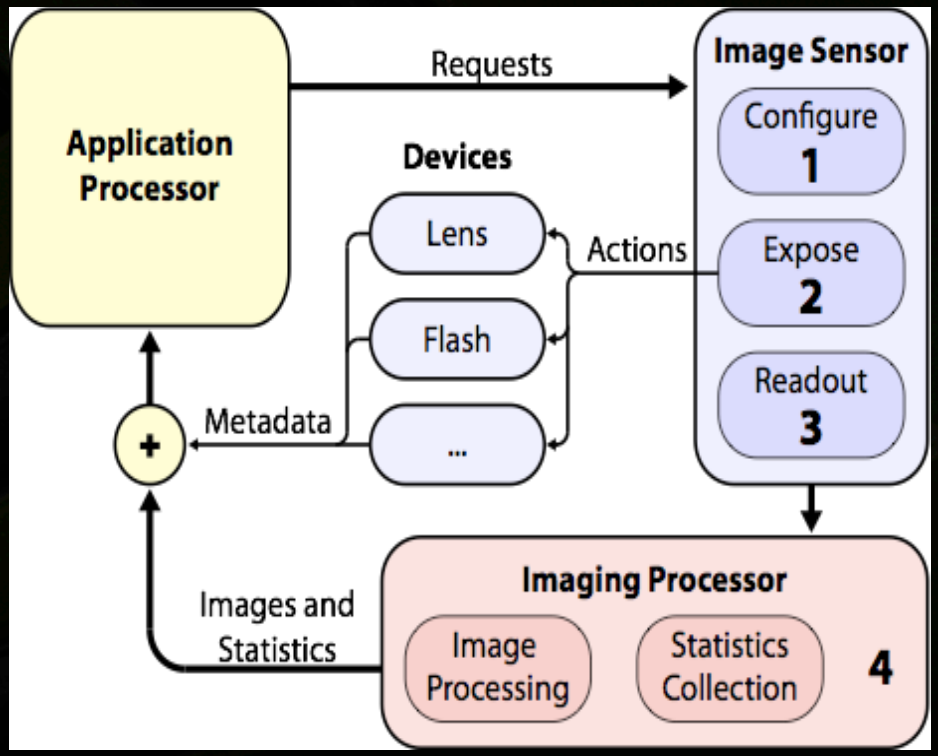


Basic Android camera is limited

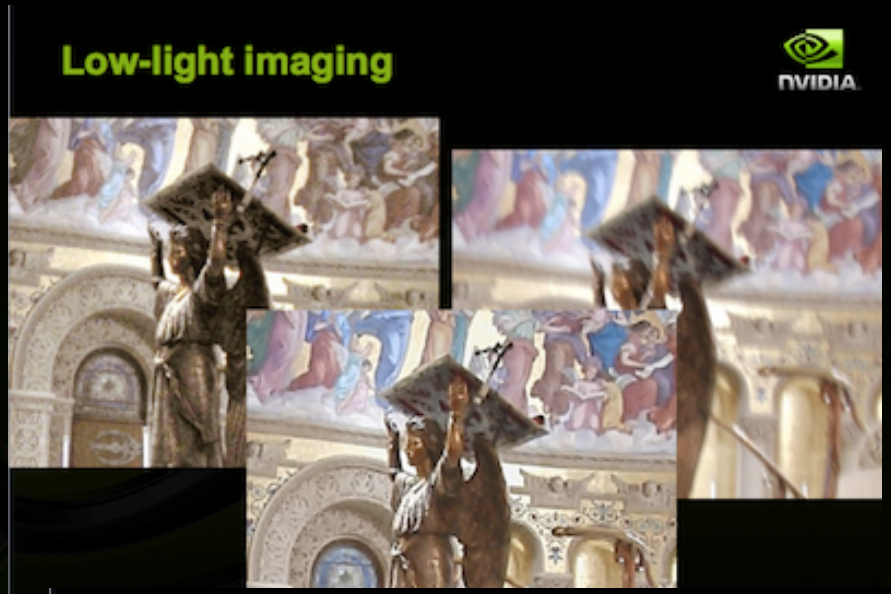
- Android provides an easy-to-use camera API
  - most low-level functionality is automated and hidden
- Not sufficient if you need accurate camera control
  - computational photography
    - quickly taking images with different settings, combine to better pictures
  - many computer vision tasks
    - the system tries to optimize the image overall, without understanding the task
    - e.g., task: recognize a register plate of a car
      - the parameters should be set so that the plate looks good, other parts of the image can look arbitrarily bad

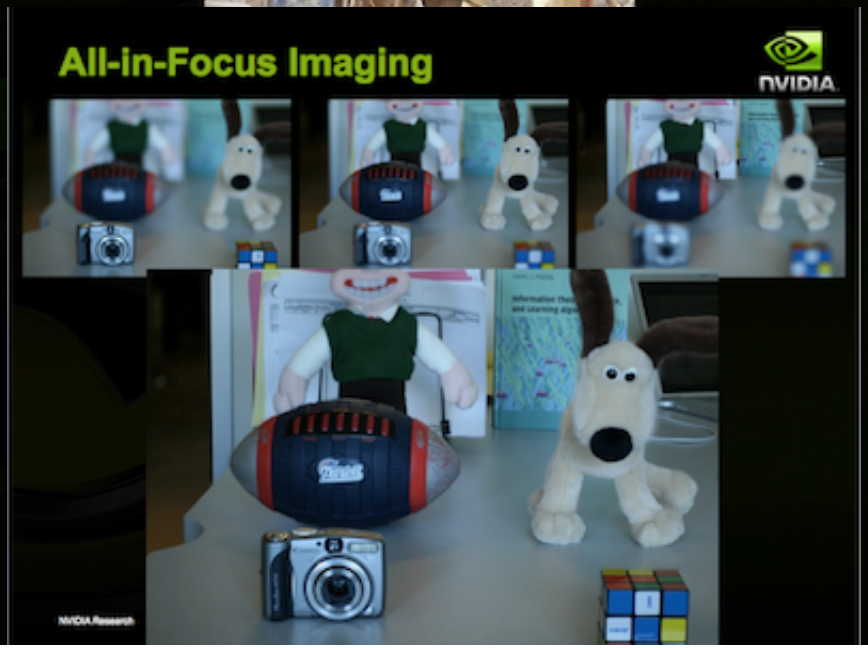
The FCam Architecture

- A software architecture for programmable cameras
  - that attempts to expose the maximum device capabilities
  - while remaining easy to program (C++)
  - modeling sensor as a pipeline allows rapid state changes



Applications





FCam: Open Source Project