Differences Between Desktop and Mobile

For now

- Do a little research on
  - The differences between desktop devices and mobile devices
  - See what you can come up with in 15 minutes.

Differences Between Desktop and Mobile

- Most of you will be used to developing for desktop environments
  - Highly likely that you have not developed for mobile before
  - There are significant differences between both platforms
  - They will be explored in this lecture

Differences Between Desktop and Mobile

- The differences include the following.
  - CPU power and architecture
  - GPU power and architecture
  - Storage space and speed

Differences Between Desktop and Mobile

- Screen size and resolution and density
- Interaction between user and device
- Power source

- All of these differences must be accounted for when you are developing applications

CPU power and architecture

- As you are coming from a desktop environment you are used to dealing with high spec processors
- e.g. the Intel Core series
  - A Sandy Bridge Core i7 processor from 2011 has a transistor count of approximately 2.27 billion.
  - That's a lot of transistors

CPU Power and architecture

- Also a lot of power consumption approx (60 to 90 watts)
- And a lot heat generated. Use of active cooling is standard with these.

- By comparison Apple's A8 ARM based processor is a complete system (CPU/GPU/and other components) has about 1 billion transistors max
  - And must use very little power and passive cooling.

CPU Power and Architecture

- This raises two important points
  - 1) you are used to programming on X86 based processors.
  - 2) you will have a significantly less computation power available.
- We will discuss these in a bit more detail

CPU Power and Architecture

- X86 processors were and still are built for speed and performance
  - Architecture design and improvements are there to speed up the execution of instructions
- ARM by contrast is built for power efficiency
  - Architecture design and improvements are there to reduce overall power consumption.

CPU Power and Architecture

- Thus when developing on mobile
  - There is a much stronger emphasis on efficiency than on desktop
  - ARM (or Intel Atom/AMD Geode for that matter) will not execute instructions as quickly as an X86 machine.
  - Anywhere you see an opportunity for increasing efficiency take it

GPU Power and Architecture

- Similar story here to CPU
  - Desktop GPU transistor counts on some of the latest processors range from 3.5 to 7 billion transistors.
  - On mobile to save on packaging and interconnects the GPU is included in the same packaging as the CPU
    - To make a SoC processor (System on Chip)

GPU Power and Architecture

- While the general architecture remains the same
  - You have a much smaller quantity of GPU render units available
  - They are also less powerful than those found in desktop
- Thus you will not be able to render as much on mobile as you can on desktop

Storage Space and Speed

- Another significant difference comes in the form of storage space
  - In the desktop world there are large magnetic hard drives (up to 4TB) and solid state storage is starting to take hold here as well (up to 1TB)
- Compared to mobile
  - All solid state storage with capacities ranging from 8 to 64GB

Storage Space and Speed

- Thus the amount of data that you can store on mobile is significantly reduced
  - Not everyone will have the latest device so assume that people have very little space available for your applications
  - Only include necessary code
  - Don't bloat your application with unnecessary features

Storage Space and Speed

- Storage speed has significant differences too
  - Standard HDDs on desktop can manage about 70 to 80 MB/s consistently
  - SSDs regularly get in excess of 300MB/s
  - NVMe can get in excess of 2GB/s
- Reading/Writing and random access of data is very quick

Storage Space and Speed

- Whereas on mobile
  - Storage is designed for reliability over speed
  - Much slower to read/write
  - However, will easily last the lifetime of the device

Storage Space and Speed

- And depending on the device your app is running on you may have to deal with internal and external storage
  - Initial implementations of Android permitted an internal flash based storage area that would be hard wired into the device
- And also a removable external storage area in the form of a MicroSD card
  - As flash storage was much more expensive at the time.

Storage Space and Speed

- Internal storage was designated for Android OS and also Applications
  - While External storage was for storing files/photos/music etc
- To give you an example of the difficulties this approach caused I'll give you a history of my last two android devices
- First device I had was a HTC Desire (Android 2.2)

Storage Space and Speed

- After the Android OS there was only 140MB of internal space remaining for applications
  - Limited to about 10 applications if you were lucky
- Applications could only be partially moved to external storage
  - If the application permitted it.

Storage Space and Speed

- My second device was a HTC One X with 32GB of internal memory
  - Only trouble was is that it was partitioned into internal and external storage
  - As android was still demanding both
- Android OS and applications were given about 6GB of the 32

Storage Space and Speed

- Thus it is still possible to get warnings about running out of space even if you have 10GB of storage remaining.
  - Thankfully this situation was recognised as a disaster
  - Android now permits a single partition to function as both internal and external storage
  - By using a few FS permissions and ACLs

Storage Space and Speed

- A lot of similar devices are still in use to this day
  - Thus when you are building applications use as little space as possible.
  - Try to reduce the number of audio/image/video files that your application uses (will be put into internal storage unless explictly declared otherwise)
  - Minimise the amount of reading and writing your application will perform.

Screen Size, Resolution and Density

- This is the biggest difference that you will face between desktop and mobile
  - There are a significant range of screen sizes (most devices including tablets fall between 4 and 10 inches in size)
  - There is also a significant range in screen resolution (at worst 800x480 to 2560x1600)
  - Thus your application has to be completely adaptable to difference screen sizes

Screen Size, Resolution, and Density

- And to complicate matters further all applications are full screen by default
  - In desktop applications you are used to having a resizable window available to you at all times
  - And also the ability to overlap windows on on top of the other
  - Android and iOS permit neither.

Screen Size, Resolution, and Density

- All of this means that your applications must be adaptable
  - They should be able to adjust their UI's depending on the size of the screen and the available resolution
  - Unforunately if you are using images in your applications you will need multiple copies to account for different display densities
    - As resizing images on a device is an expensive image processing task especially if you require high quality resizing.

Screen Size, Resolution, and Density

- Even though some devices have large screens (Nexus 6 has 2560x1600 for a 6 inch screen !!!)
  - This does not necessarily mean you can feasably pack in more and more data into a smaller space
  - Applications must still be readable and usable
  - Must decide carefully what is the most important information that must be shown to users.

Screen Size, Resoluton, and Density

- Also if your application can run on a much larger device (e.g. a tablet)
  - It should be able to adjust the UI to take advantage of the **physical** space
  - A good example of this example of this is to compare the difference between the Uis of Facebook and Google+ on a tablet.
    - Google+ makes two columns of posts compared to Facebook's overblown post size on a single column

Screen Size, Resolution, and Density

- Thus your applications must take account of physical screen size first
  - Then display density
  - Your app should be designed to work well on a 4-6 inch screen first
  - Then design an interface for tablets (7-10 inches) afterwards

Interaction Between User and Device

- Another significant difference between Desktop and Mobile is user interaction
  - On desktop user's generally use a pointer guided by a mouse, trackpad, or trackball
  - Pointers offer single pixel precision
    - Therefore controls can be smaller and control density can be increased to make smaller windows

Interaction Between User and Device

- Whereas on mobile user interaction is done through touch
  - The difference between touch and a pointer is that touch represents an area of pixels (roughly circular) whereas a pointer represents a single pixel
  - Thus touch is not as precise.
  - This means that all of your controls must be larger to reduce user input error.

Interaction Between User and Device

- Thus your control density is reduced
- If you couple this with the fact that your screen size is also reduced
  - Your UI as a whole has very little space to display information
  - Meaning you must consider very carefully if each control or piece of information is necessary

Interaction Between User and Device

- Controls must also be large enough for users to comfortably interact with.

Power Source

- Another significant difference that will affect your application design is the difference in power source
  - Desktop machines are plugged into AC power so there is no concern over how much power is available
  - Mobile devices by contrast are limited by battery
  - Battery technology has not advanced as quickly as CPU technology over the last 20 years

Power Source

- Most devices can manage about a day on battery with some careful management from the user
  - Android also keeps full stats on what applications are consuming the battery the most
  - If your application is inefficient and wastes battery it will be removed quickly and an alternative will be found
  - Thus your applications must be as efficient as possible with respect to battery usage.

Power Source

- Every action that you take in your application will cost battery power
  - Powers screen/CPU/GPU everything
  - The less of this you use the better