Blink & Mobile

Peter Beverloo
<peter@chromium.org>

Questions?

Either ask them while we’re at the slide,
or add them to the Moderator:

http://goo.gl/9RK6Vl

Blink on Android

Different from other platforms:

- Cross-compiles to the target platform on Linux *(ARM/x86)*
- Platform implementation often written in Java *(content/)*
  - .. uses JNI for Java ←→ C++ communication

- Builds .so libraries, then packages an APK
- Runs on a device, uses “adb” for communication
  - .. most test suites require rooted devices.

Blink on Android

Subject to various constraints:

- Less memory, less CPU power, worse GPU.
  - .. yet overall large screen resolutions.
  - .. yet users expect fast startup, loading and 60fps, accurate and immediate touch input, and so on.

- Slower flash drives; can ship on ARM and x86.

Memory constraints

- Modern phones usually have > 1GB memory.
    - .. memory will often be shared with VRAM.
    - .. not everyone has a modern phone. 512 megs?

- Using a lot of memory? Expect to be killed.
    - .. system can kill the application at any moment.
    - .. *sometimes* it’s nice and gives you a heads-up.

- Be careful with caching data (purge them, share them).

CPU constraints

- Introduce All The Threads!
  - .. works great on powerful desktop machines.
  - .. *may* work great on Android.
    - .. 2 cores and 20 active threads?

- Measure! Don’t assume it’ll be faster because you *think* it’ll do more at once. The CPU may not be able to cope.

- No cooling fan in the devices. Is the CPU getting too hot? Let’s tune it down a little bit.

Other mobile constraints

- Worse GPU, yet at least equally large displays.
  - .. threaded painting and scrolling.
  - .. check out the Graphics talk by Nat and Tom.

- Need to be careful about power usage.
- Network is unreliable and costly.

Opportunities!

Not everything about mobile is worse!

- Big growth in number of users.
- New device generations bring many improvements.
- Lots of opportunities for improving UX.

- Improving memory and performance on mobile usually means improving it elsewhere as well.

How to build Content Shell

1. Modify your .gclient file to include

   ```
   target_os=[‘android’]
   ```

2.
   ```
   $ gclient sync
   ```

3.
   ```
   $ . build/android/envsetup.sh
   $ android_gyp

   $ ninja -c out/Release content_shell_apk
   ```

How to run Content Shell / tests?

Running content_shell:

-
  ```
  $ build/android/adb_install_apk.py --apk ContentShell.apk
  $ build/android/adb_run_content_shell
  ```

Running layout tests:

-
  ```
  $ webkit/tools/layout_tests/run_webkit_tests.py
        --android
  ```

WebKit unit tests

-
  ```
  $ build/android/test_runner.py gtest -s
        webkit_unit_tests
  ```

How to run Content Shell / tests

Running the performance tests:

-
  ```
  $ third_party/WebKit/Tools/Scripts/run-perf-tests
      --android
  ```

Or use Telemetry!

- http://www.chromium.org/developers/telemetry

Notes when running tests..

- You’ll need to have at least one actual device attached.
- Right now, ~225 layout tests run on the Android bot on the Blink waterfall. But… very flaky!

Discussion!

- Questions? Comments?

http://goo.gl/9RK6Vl