[SHORT] Temat: Android Profiler - Szukanie źródła niewydajnego kodu.

<PRZYWITAĆ SIĘ I URUCHOMIĆ APLIKACJĘ>

Powiedzieć, że będą dwa przykłady: problem w kodzie kotlina i pliku XML z layoutem

<OMÓWIĆ LISTĘ I ZADEMONSTROWAĆ SPADKI PŁYNNOŚCI (COMPLEXITY=33)>

Powiedzieć, że cel to 60 FPS. Powiedzieć, że w dokumentacji jest to nazwane JANK

<URUCHOMIĆ PROFILER>

<POWIEDZIEĆ O SPOSOBACH REJESTROWANIA AKTYWNOŚCI CPU>

Sampled method tracing. Instrumented method tracing. System tracing (systrace).

<USTAWIĆ COMPLEXITY=32 I ZAREJESTROWAĆ SYSTRACE>

<POKAZAĆ DISPLAY/FRAMES I THREADS/MAIN & THREADS/RenderThread>

Symbioza wątku głównego z RenderThread.

<POKAZAĆ NAWIGACJĘ Z POMOCĄ KLAWIATURY>

<WYBRAĆ JEDNO CZERWONE ZDARZENIE I PRZYBLIŻYĆ JE>

<POKAZAĆ I OMÓWIĆ ZDARZENIE „RV OnBindView” >

<WYKONAĆ I OMÓWIĆ WIDOK METHOD TRACING>

<ZAZNACZYĆ WĄTEK GŁÓWNY, OMÓWIĆ CALL CHART I TOP DOWN>

<WPISAĆ „onbindview” W TOP DOWN I POKAZAĆ TO NA WYKRESIE WYWOŁAŃ>

<SKOCZYĆ DO KODU>

<UŻYĆ LEPSZEJ IMPLEMENTACJI I POKAZAĆ RÓŻNICE>

<POKAZAĆ „LL SCREEN”>

<WYKONAĆ SYSTRACE NA LL SCREENIE I OMÓWIĆ ZDARZENIE „measure”>

<WYKONAĆ SYSTRACE NA CL SCREENIE I OMÓWIĆ>

<BONUS: WIZUALNA IDENTYFIKACJA JANKÓW. POKAZAĆ HISTOGRAM UŻYCIA GPU>

Oznaczenie kolorów na histogramie:

- Pomarańczowy -> Reprezentuje czas spędzony przez CPU na czekaniu na GPU aż skończy pracę. Jeśli słupek jest wysoki to aplikacja wykonuje za dużo pracy na GPU.

- Czerwony -> Reprezentuje czas spędzony przez moduł renderujący 2D systemu Android na wysyłanie poleceń do OpenGL.

- Jasny Niebieski -> Reprezentuje czas spędzony na ładowaniu bitmap do GPU.

- Ciemny Niebieski -> Reprezentuje czas spędzony na tworzeniu i aktualizacji widoków. Jeśli ta część jest wysoka to oznacza, że aplikacja wykonuje dużo rysowania niestandardowych widoków w metodach \*.onDraw(...).

- Najjaśniejszy Zielony -> Represents the amount of time spent on onLayout and onMeasure callbacks in the view hierarchy. A large segment indicates that the view hierarchy is taking a long time to process.

- Średni Zielony -> Reprezentuje czas potrzebny do oceny wszystkich animatorów uruchomionych dla tej ramki i obsługi wszystkich wywołań zwrotnych wejściowych. Jeśli ten segment jest duży, może to oznaczać, że niestandardowy animator lub wejściowe wywołanie zwrotne spędza zbyt dużo czasu na przetwarzaniu. Wiązanie widoku podczas przewijania, takie jak RecyclerView.Adapter.onBindViewHolder(), również zwykle występuje podczas tego segmentu i jest częstszym źródłem spowolnień w tym segmencie.

- Najciemniejszy Zielony -> Reprezentuje czas, jaki aplikacja spędza na wykonywaniu operacji między dwiema kolejnymi ramkami. Może to wskazywać na zbyt duże przetwarzanie w wątku interfejsu użytkownika, które można przenieść do innego wątku.