



Contribution ID: 42

Type: **not specified**

Improving Graphics Interactivity - It's All in the Timing

Tuesday, 13 November 2018 09:00 (45 minutes)

Interactive applications, which includes everything from real time games through flight simulators and virtual reality environments, place strong real-time requirements on the whole computing environment to ensure that the correct data are presented to the user at the correct time. This requires two things; the first is that the time when the information will be displayed be known to the application so that the correct contents can be computed, and second that the frame actually be displayed at that time.

These two pieces of information are managed inconsistently through the graphics stack, making it difficult for applications to provide a smooth animation experience to users. And because of the many APIs which lie between the application rendering using OpenGL or Vulkan and the underlying hardware, a failure to correctly handle things at any point along the chain will result in stuttering results.

Fixing this requires changes throughout the system, from making the kernel provide better control and information about the queuing and presentation of images through changes in composited window systems to ensure that images are displayed at the target time, and that the actual time of presentation is reported back to applications and finally additions to rendering APIs like Vulkan to expose control over image presentation times and feedback about when images ended up being visible to the user.

This presentation will first demonstrate the effects of poor display timing support inherent in the current graphics stack, talk about the different solutions required at each level of the system and finally show the working system.

I agree to abide by the anti-harassment policy

Yes

Primary author: PACKARD, Keith (Hewlett Packard Enterprise)

Presenter: PACKARD, Keith (Hewlett Packard Enterprise)

Session Classification: LPC Main Track

Track Classification: Refereed talk