After the wireframing and prototyping phases, you can use previewing and profiling tools to fine-tune your UI for production.

How to achieve production quality motion in UIs:

- Preview the UI to check the FPS refresh rate.
- Profile the UI code to find causes for slowness, unresponsiveness, and stuttering.

FPS Refresh Rate

As a general rule, animators strive to allow the rendering engine to achieve a consistent 60 frames-per-second (FPS) refresh rate. 60 FPS means that there is approximately 16 milliseconds between each frame in which processing can be done, which includes the processing required to upload the draw primitives to the graphics hardware.

The frames-per-second (FPS) refresh rate of animations is displayed in the **FPS** field on the toolbar in the **Design** mode.

To improve the FPS rate, application developers should:

- Use asynchronous, event-driven programming wherever possible.
- Use worker threads to do significant processing.
- Never manually spin the event loop.
- Never spend more than a couple of milliseconds per frame within blocking functions to avoid skipped frames, which negatively affect the user experience.

For more information about previewing UIs on devices, see Validating with Target Hardware.

Profiling UI Code

You can use QML Profiler that is integrated into Qt Design Studio to find causes for typical performance problems in your UI. For example, your UI might be slow, unresponsive, or stuttering. Typically, such problems are caused by executing too much JavaScript in too few frames. All JavaScript must return before the GUI thread can proceed, and frames are delayed or dropped if the GUI thread is not ready.

In general, knowing where time is spent in a UI enables you to focus on problem areas that actually exist, rather than problem areas that potentially exist.

Determining which bindings are being run the most often or which functions your application is spending the most time on enables you to decide whether you need to optimize the problem areas, or redesign some implementation details of your application so that the performance is improved. Attempting to optimize code without profiling is likely to result in very minor rather than significant performance improvements.

For more information, see Profiling QML Applications.

‹ Editing Easing Curves   Optimizing Designs ›