Subject: Bézier Curve and animation Engine for U++ CoreLib
Posted by dodobar on Thu, 04 Sep 2025 06:10:09 GMT
View Forum Message <> Reply to Message

As per the get hub discussion:
https://github.com/orgs/ultimatepp/discussions/274

(https://github.com/Trilec/upp_AnimationEasing)

Quick update after a few weeks on this.
The little easing header I dropped earlier has grown into a proper animation lib.

What's working now
• Core ops: play, cancel, stop, delay, looping/yoyo, pause/resume.
• Easing: both presets and custom Beziers.
• Safe: reentrancy is handled, memory cleans up properly (no leaks or dangling refs).
• Stress tests: concurrent bursts, pause/resume loops, cancel/restart abuse... holding up well so far.
• API surface is still U++-native and clean (AnimateProperty, AnimateFade, AnimateColor, Easing::Bezier, etc.).

What's different from the old helpers
• Old Animate() was narrow and ad-hoc; this lib has one consistent core + thin wrappers.
• Wrappers (Fade, Rotate, Pulse, etc.) are now just sugar over the same engine.
• Animations finish/cancel deterministically -- no dangling timers.
• Designed with UI integration in mind rather than bolted on.

Next steps
Stress tests + examples DONE , PR-ready.

Integration concepts (small + safe):

Keep runtime self-contained (Animation.h/.cpp).
Add one light layer in Chameleon: an animated look wrapper that interpolates between normal/hot/pushed/disabled. Themes can opt-in by swapping style.look[] to this animated one.
Same helper can animate draw-time values (colors, fonts, underlines, sliding panels).
Global switch ChAnim::SetEnabled(bool) for users who want to disable animations.

This keeps it host-look-first, opt-in, and widget code stays untouched. A couple of tasteful demos (button hover fade, focus ring pulse, tab highlight slide) would prove the path.

Would love feedback on whether this direction feels right before I wire it into CtrlLib.
examples of integration so can begin testing

UI TEST Demo :

File Attachments

1) animation.jpg, downloaded 40 times