MetaHuman Steering Streamer GUI

Working Prototype - Real-time OSC Control System

EXECUTIVE SUMMARY

Successfully developed a working desktop GUI prototype that enables real-time control of MetaHuman characters in Unreal Engine through OSC (Open Sound Control) messaging. The system provides intuitive steering controls with smooth animations and real-time connection monitoring.

KEY FEATURES & CAPABILITIES

Feature	Description	Status
Real-time OSC Streaming	60 FPS continuous data streaming to Unreal	■ Working
Multi-channel Control	16 synchronized steering channels	■ Working
Smooth Animations	Cubic ease-in-out ramping system	■ Working
Connection Monitoring	Live status indicator & error tracking	■ Working
Configurable Parameters	IP, Port, FPS, Duration, Hold settings	■ Working
Keyboard Shortcuts	Hotkeys for rapid control (Ctrl+S, R, L, B, X)	■ Working
Settings Persistence	Auto-saves configuration to user profile	■ Working
Error Handling	Graceful error recovery & logging	■ Working

TECHNICAL IMPLEMENTATION

Component	Technology	Purpose
GUI Framework	Tkinter (Python)	Cross-platform desktop interface
OSC Communication	python-osc library	Real-time data streaming
Animation Engine	NumPy + custom easing	Smooth motion curves
Threading	Python threading	Non-blocking GUI + background processing
Configuration	JSON format	Flexible channel mapping
Error Handling	Exception management	Robust error recovery

BUSINESS VALUE & IMPACT

- **Rapid Prototyping:** Enables quick testing of MetaHuman steering concepts
- **Real-time Control:** Provides immediate feedback for animation development
- **Scalable Architecture:** Easy to extend with additional channels or features
- **User-Friendly Interface:** Intuitive controls accessible to non-technical users
- **Production Ready:** Robust error handling and connection monitoring
- **Cost Effective:** Single-file solution with minimal dependencies

NEXT STEPS & RECOMMENDATIONS

The prototype is ready for integration testing with Unreal Engine MetaHuman projects. Recommended next phase includes user acceptance testing, performance optimization, and potential integration with existing motion capture workflows.