



MacroMancer Product Review *(First Draft)*

Project Update and Overview

Presented by: Alexander Shelton

Project Overview

- ◆ MacroMancer is a customizable tool that enables Linux users to record and bind different actions or commands to keyboard and mouse actions.
- ◆ **Key Benefits:**
 - ◆ Enhances workflow automation
 - ◆ Supports custom key bindings and macros
 - ◆ Designed for flexibility and expandability

Core Features

- ◇ Capture and record keyboard and mouse inputs
- ◇ Save recorded input and user configurations in a JSON/YAML file
- ◇ Interactive command-line interface (CLI)
- ◇ Configuration file for user customization
- ◇ Modular architecture for future expansions

Requirements

- ◆ Ability to record and store bindings for keyboard and mouse inputs
- ◆ User-defined triggers and actions
- ◆ Lightweight design for Linux users
- ◆ Easy integration of additional features in future iterations

Project Deliverables



A functional CLI application for recording and managing macros



JSON/YAML-based configuration and storage system



Modular design for extensibility



User-friendly documentation and examples

Project Status

Current Phase

Testing Phase

Major Design Work

Completed

Core Functionality

Operational

Overall Progress

80%

Challenges & Adjustments

◆ **Challenges Faced:**

- ◆ Managing scope to meet deadlines
- ◆ Recording and replaying long macros
- ◆ Mapping `pynput` keys to the appropriate keycodes.

◆ **Scope Adjustments:**

- ◆ Postponed more advanced macro recording and execution to next iteration
- ◆ Focused on refining current features and stability

Risk Assessment

◆ Identified Risks:

- ◆ Inconsistencies in input capture module on different types of keyboards and mice
- ◆ Inconsistent behavior across different Linux distros and window managers/desktop environments

◆ Mitigation Strategies:

- ◆ Focus on core functionality in this iteration
- ◆ Focus on one desktop environment before expanding
- ◆ Allocate additional time for testing

Project Progress - WBS

- ◆ **Completed Tasks:**

- ◆ Capture of keyboard and mouse input
- ◆ JSON/YAML configuration integration
- ◆ Basic CLI functionality
- ◆ Binding commands to user defined actions

- ◆ **Ongoing Tasks:**

- ◆ Extensive testing and debugging
- ◆ Help documentation and examples

Remaining Items

- ◇ Complete user documentation
- ◇ Conduct testing
- ◇ Gather feedback for refinements
- ◇ Enhance CLI usability
- ◇ Address minor bugs and issues

Future Tasks (Backlog)

- ◇ Improve UI/UX for the CLI interface
- ◇ Conduct performance benchmarking
- ◇ Expand scope to include advanced macro recording and scripting.

Questions & Feedback

- ◇ We appreciate your support and insights!
- ◇ Please share your feedback on:
 - ◇ Functionality
 - ◇ Usability
 - ◇ Potential Improvements
- ◇ Thank you for your time!