An Introduction to SLEWN:A Macro Language for Sky AutoTrack

1 Why Macros?

In the spirit of making astronomy more accessible to the novice and more streamlined for the expert, macros were added to this system to allow the user to automate repetitive tasks. We decided on macros instead of a full fledged programming language in order to cut down on the syntax and nuances a user would have to learn. Because of this, the macros system has no variables and only accepts simple, procedural commands.

2 What Can I Do?

The name SLEWN reveals everything our language can do: Slewing, Locating, Exploring, Waiting, Interpreted language

1. **Slew:** You can send commands for the telescope to slew to specified coordinates.
2. **Record:** You can demarcate the begin and end of a sequence of actions to record.
3. **Park:** You can send the telescope to its designated home position.

3 Getting Started

First, it's good to know how to write the simplest (and most useless) macro available to this language. As **Figure 1** shows, each macro begins with "Begin" and ends with "End".

```
Begin

End
```

Figure 1: A Useful Example

All of the interesting stuff comes between those two words. In order to discuss these more interesting features, let's look at a more complex example such as **Figure 2**.

Here we can see recording in action. Every statement between "Record" and "Done" will be executed while a recording window is up and running.

Slew behaves just as expected, with the coordinates being accepted as floating point numbers with arbitrary precision.

Note: Whitespace does not matter in this language, but tabs are encouraged for readability.

```
Begin
    Record;
        Slew to (3.0, 45.0);
        Park;
        Slew to (12.5, 16.35);
    Done;
    Park;
End
```

Figure 2: A More Useful Example

4 Writing Macros

In order to begin writing and running your macros, simply select "File" from the main application page as shown in **Figure 3**. From here you can select to open a new file, open a previously written file, or open and simultaneously run a previously written macro.

The editor for the macro is entirely self contained in our application, but you can open a plain text file filled with your code from any source you want.

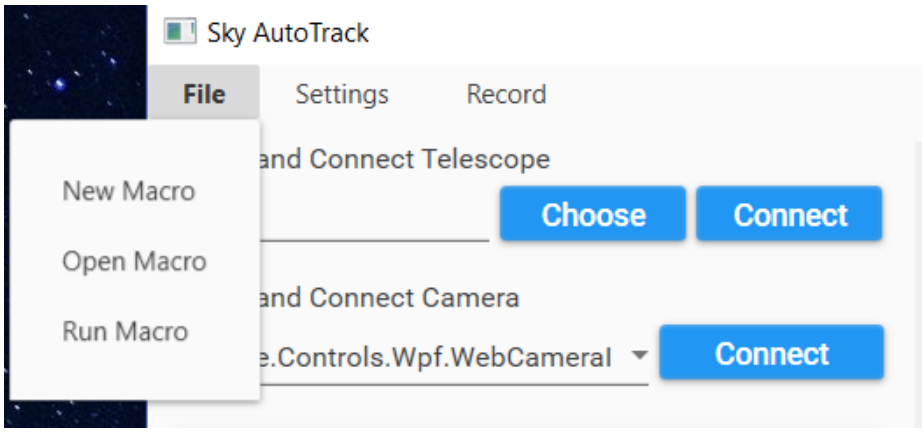

Figure 3: Macro Creation/Selection

5 Executing Macros

After you have written a syntactically valid macro, simply press the "Run" button in the upper right hand corner of the editor to hand your program off to the interpreter to run your program.

6 Errors

The status of the editor can be seen in the lower left hand corner of the editor window. Under normal conditions, this status bar will display "OK" as shown in **Figure 4**. If you attempt to run your macro without a connected telescope you will see an error signal as shown in **Figure 5**. If you forget a semicolon or use an unexpected command, the interpreter will provide you the error as shown in **Figure 6**.

Figure 4: Macro Status Bar: "OK"

Figure 5: Macro Status Bar with No Telescope Connected

Figure 6: Macro Status Bar with Syntax Error