**Last Stop Development Journal**

|  |  |
| --- | --- |
| Wed March 11 | 4:30 PM. Development begins. I’m debating between using the .NET Framework in C# and between using Electron/TypeScript. After a bit of research, I conclude that using C# is the more prudent decision. It feels like this project would benefit from a more robust type system, and Dragon is able to use its full feature set with Windows Forms. Otherwise, I’d have to do RPC with a WinForms stub.  5:30 PM. First check in. Titled the project “Last Stop” because I hope this is the last time I’ll ever try to make an editor. Feeling discouraged, low energy. I’ve been here before and it doesn’t feel good to start from zero again. This really has to be the last time.  Decision paralysis sets in. Not sure where to start, what the project scope should be. The ultimate goal is to be able to make software by voice with no compromises in any language (provided I’ve made voice definitions for it). Many ways to achieve that goal, all different.  Preliminary structures:   * Model: a data object containing all reversible (i.e. voice controlled) information for the application. The model is only modified by deltas. Snapshots can be taken of the model (these are stored OUTSIDE the serialized model). Undo commands amount to re-applying stored deltas to a recently taken snapshot. (This prevents the need for storing inverse deltas, which is really a pain in the butt.) * External: an object containing all information which is not reversible, that is, outside of voice control. Things like changes on the file system, time/date, network access, and so on belong here. * View: The collection of windows which subscribe to the model. * Windows: each window subscribes to a certain view of the model. Changes in the model trigger events sent to the window. contains a single UserControl and subscribes to a certain view of the model. * Content: a character-matrix UserControl with background/color/in-text markup options * Input: a textbox UserControl |
|  |  |