The Concept

The idea of drift was a “racing game”. Not necessarily against AI, but a game that had a car sprite and track playfield. I wanted a game whose gameplay was baked into its essence. Given my inexperience programming in assembly, I wanted to give myself a creative and interesting playfield to work with. I ended up working with an application on atariage. This converts images (png, jpeg, etc) into 6 playfield columns and bit tables. This took some time because it required the use of photoshop. Sparing the details, I successfully created a playfield using this application. I used a one-line kernel to produce a zoomed-out effect. Whenever a user’s sprite enters the playfield, it slows down drastically or completely stops, depending on which way they’re going.

The execution of drifting

While messing with player movement, I was looking for a way to create inertia/momentum. Keeping the HMP0 rather than storing 0 into it after each frame was my solution. This forces the user to constantly change between up/down and right/left in order to lap around the playfield. User timing determines the likelihood of whether their car will be caught on the playfield. A part of the draw in this game is the satisfaction in making well-timed drifts.