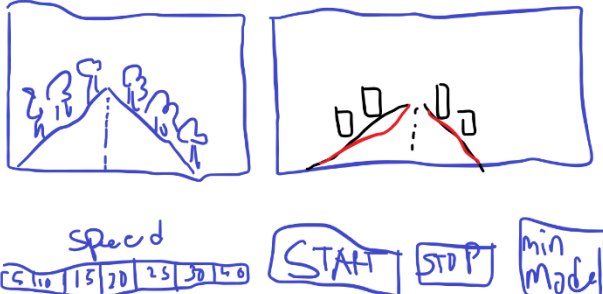
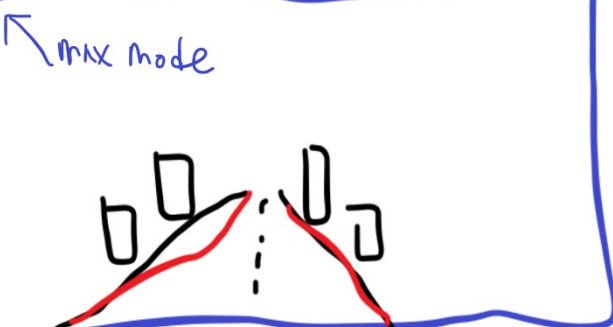



<p>Jalopy</p> <p>Preparing Insc...</p>	<p>Jalopy by Eric Chang</p> <p>Play</p> <p>Help</p>	<p>Help</p> <p>Overview</p> <p>Setting up game</p>
<p>Loading screen: I first dazzle the user with cheeky loading messages as this program initializes on the user's computer. This is also time for the user to have Euro Truck Simulator 2 running in the background, ready to go.</p>	<p>Start screen: The user can begin <i>Jalopy</i>, or can view a help screen with information on how to best run the game for ease-of-use and performance. The right hand side of the screen would be an animated gif of a rotating truck (taken from in-game)</p>	<p>Help screen: Information on how to run Euro Truck Simulator 2, including tips on graphical settings, key bindings and sensitivities, and other settings</p>
		
<p>Primary program screen: The game input is captured at n-fps on the left hand side of the screen, while the processed image is handled on the right (with lane detection, boxes for detectable objects i.e. trees, pedestrians, cars).</p> <p>The user can manipulate the target speed modifier of the truck, as well as make emergency stops.</p>	<p>Minimized program screen: Clicking the 'min-mode' button on the lower right hand corner of the primary program screen allows the user to admire the pitch-black, soulless landscape of <i>Jalopy</i>'s computer vision work.</p>	<p>Literal crash screen: When <i>Jalopy</i> thinks that the user crashed in-game, it will display a warning that <i>Jalopy</i> has detected a collision and will direct the user back to the main menu, with stats on how long <i>Jalopy</i> was able to survive and estimated distance (as a crude function of time * speed)</p>
<p>Note: All screens incorporate smooth animation between frames, rather than abrupt transitions</p>		