‘

Jalopy

Algorithmic plan

Structural plan

Python plays Euro Truck Simulator 2[[1]](#footnote-1)

Version control plan

Module list

I am building a self-driving algorithm for the video game Euro Truck Simulator 2.

Project overview

Many have tried to build similar self-driving algorithms for this game; many have succeeded.

Competitive analysis

The popular open-source package for the game *europilot* by *marsauto* involves using a convolutional neural network (cNN) to direct the truck. From what I have seen from the GitHub repository, *europilot* takes several screenshots of the game’s state each second, and then runs in either one of two ways. If the user has activated ‘training mode,’ then the screenshot will be mapped to key-pressed values as a new row entry in a .csv file. This creates a ‘virtual joystick’ for the program to begin learning how the user drives.

On the other hand, if the user is in ‘testing mode,’ the virtual joystick will output the relevant commands to the truck, such as steering, shifting gears as necessary, accelerating, and braking. It uses the training data accumulated from testing mode to generate its inferences on how to best direct the vehicle.

I envision *Jalopy* will be like *europilot* and many other packages with its usage of OpenCV to track game state. However, I am not certain yet if I will use a neural network in my implementation.

Timeline plan

Algorithmic plan

1. Euro Truck Simulator 2 (ETS2), published by SCS Software, is a popular 2012 driving simulator video game for Microsoft Windows, Mac OS, and Linux. Players drive and deliver cargo across Europe.­­ [↑](#footnote-ref-1)