Dev Zaveri, Tyler McCluskey, Ken Chambers, Navid Hoque

CMPSC 470 Sections 001, 001, 002, 001 [in order of names]

Nelson

29 April 2024

**Car Interpreter Language Whitepaper**

**Abstract**

Car Interpreter Language (CIL) is a domain-specific language designed for controlling simulated car functions in a user-friendly and intuitive manner. This whitepaper provides an overview of the Car Interpreter Language, including its syntax, features, applications, and benefits. It also discusses the architecture of the Car Interpreter Language and provides examples of its usage.

**1. Introduction**

In the field of automotive simulation and testing, there is a need for a simple yet powerful scripting language to control simulated car functions. CIL is specifically designed to address this need by providing lightweight and intuitive language for defining car behaviors and interactions.

**2. Syntax and Features**

**2.1. Commands**

CIL supports a set of commands for controlling simulated car functions, including:

* **drive()**: Initiates forward motion of the car.
* **stop()**: Stops the car's motion.
* **reverse()**: Engages the car in reverse motion.
* **horn()**: Activates the car's horn.
* **read\_sensor()**: Simulates reading data from sensors, such as obstacle detection.

**2.2. Control Structures**

CIL includes control structures such as:

* Conditional statements (**if**, **else**): Allows for conditional execution of code blocks.
* Loops (**while**): Enables repetitive execution of code blocks based on specified conditions.

**2.3. Arithmetic Operations**

CIL supports basic arithmetic operations for performing calculations and storing results in variables.

**3. Applications**

CIL is suitable for a variety of applications in simulated car control and testing, including:

* Autonomous vehicle simulation: Simulate autonomous driving algorithms and evaluate their performance.
* Educational purposes: Introduce students to concepts of car control and programming in a simplified environment.

**4. Benefits**

**4.1. Simplicity**

CIL’s simple syntax and intuitive commands make it easy for users to define car behaviors and interactions without needing extensive programming knowledge.

**4.2. Flexibility**

The combination of control structures and arithmetic operations provides flexibility in defining complex car control logic and scenarios.

**4.3. Integration**

CIL can be easily integrated into existing simulation frameworks and tools, allowing for seamless incorporation into simulation environments.

**5. Interpreter Architecture**

The CIL interpreter translates Javascript code into executable commands for simulated car functions. It consists of components for tokenization, parsing, and execution, as well as an input processor for obtaining CIL code input.

**6. Conclusion**

CIL provides a simple yet powerful solution for controlling simulated car functions in automotive simulation and testing. Its intuitive syntax, flexible features, and ease of integration make it an ideal choice for developers, researchers, and educators working in the field of automotive simulation.