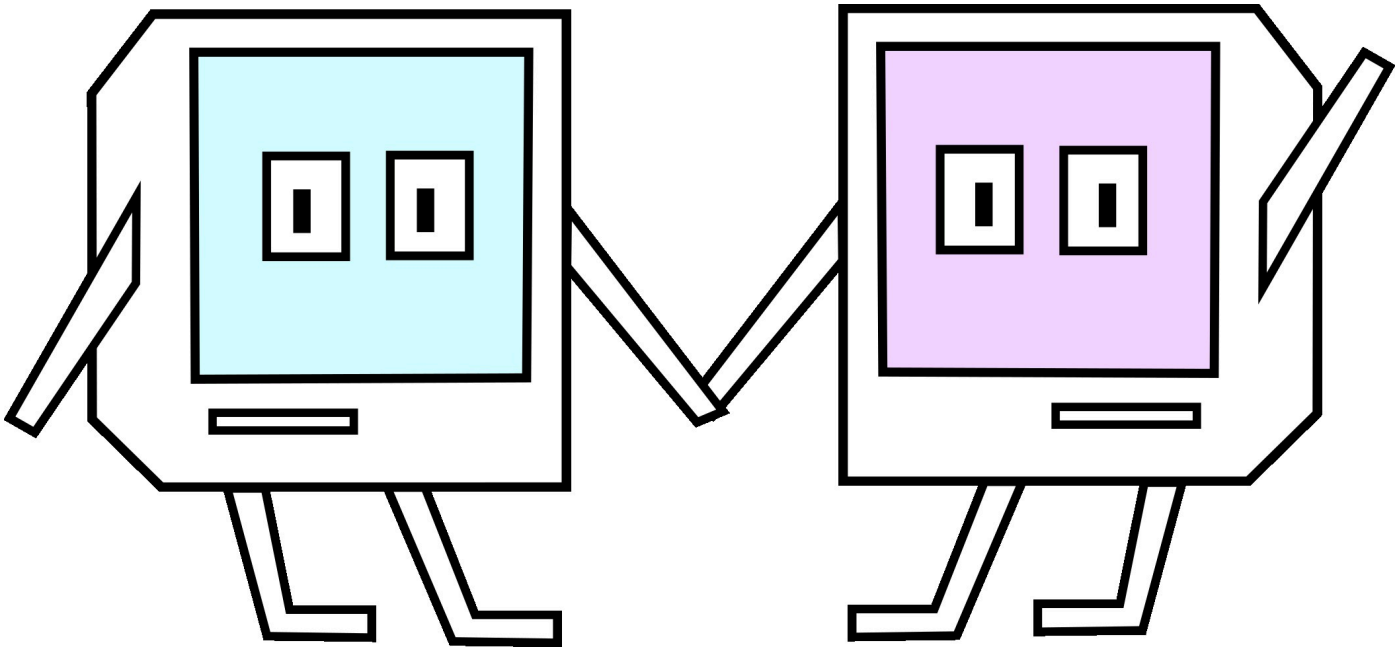

Gismo Programming Language

This is the repository for the **Gismo Programming Language**—a meta-programming language that allows you to extend its syntax and functionality as needed. Gismo combines the extensibility of Lisp with a C-like syntax, making it both powerful and familiar.

🚀 Overview

Gismo is designed for meta-programming, enabling developers to create and extend the language with new syntaxes and constructs. Instead of a traditional compiler, Gismo uses an **Interpiler**—a hybrid of an interpreter and a compiler—that can be programmed to generate code for any backend.

Key Concepts

- **Extensible Syntax**: Define or overload operators and create new language constructs.
- **Interpiler Architecture**: Program the interpiler to generate code, offering flexibility in code generation.
- **Lisp-like Meta-Programming**: Achieve powerful meta-programming capabilities similar to Lisp but with a C-like syntax.
- **Operator Overloading & Type System**: Utilize advanced features to customize language behavior.

👥 Target Audience

Gismo is ideal for:

- **Hobbyists and Researchers**: Interested in creating custom dialects and experimenting with language design.
- **Meta-Programmers**: Developers who want to extend the language capabilities to suit specific needs.
- **General Programmers**: Once a dialect is established, Gismo can be used by anyone for various programming tasks.

✨ Features

- **Customizable Code Generators**: Write different code generators to compile to any backend.

- **Define New Operators**: Use the `::=` syntax to define or overload operators.
- **Inbuilt Functions for Language Extension**: Use functions prefixed with `$` to extend the language.

Example: Defining a New Operator

```
int + int ::= $ADD($1, $2)
```

This snippet tells the interpiler to translate every `+` operation between two integers into `$ADD($1, $2)`, where `$1` and `$2` are placeholders for the operands.

📝 "Hello, World!" Example

While there's no traditional "Hello, World!" in Gismo, you can make the interpiler output "Hello, World!" at compile time using the `$PRINTLN` inbuilt function:

```
$PRINTLN("Hello, World!")
```

🔧 Installation

Prerequisites

- **Go Language**: Ensure you have Go installed on your system.

Steps

1. **Clone the Repository**

   ```
   git clone https://github.com/your-username/gismo.git
   cd gismo
   ```

2. **Build the Interpiler**

```
go build -o compiler
```

3. **Run a Gismo Program**

```
./compiler [path to your Gismo file]
```

🌟 Getting Started

Create a file named `example.gm`:

```
int + int ::= $ADD($1, $2)
$PRINTLN("Extending Gismo!")
```

Run the interpiler:

```
./compiler example.gm
```

🛣️ Roadmap

Future plans for Gismo include:

- **Standard Library Development**: Building a comprehensive standard library.
- **Enhanced Backend Support**: Extending code generation to more backends.
- **IDE Integration**: Developing plugins for popular IDEs.
- **Debugger Tools**: Implementing debugging tools tailored for meta-programming.
- **Community Dialects**: Encouraging the creation of dialects with their own standard libraries.

🤝 Contributing

Contributions are welcome! Here's how you can help:

- **Pull Requests**: Submit PRs to add features, fix bugs, or improve documentation.
- **Issues**: Report bugs or suggest features by opening an issue.
- **Discussions**: Participate in discussions to help shape the future of Gismo.

📣 Community and Support

Stay tuned for community channels. A Discord server may be created for real-time discussions and support.

📄 License

This project is licensed under the MIT License.

💡 Feedback

We'd love to hear your thoughts on Gismo! Feel free to open an issue with any feedback, questions, or suggestions.