

CSC 402 - Final Project Proposal

Names: Danny Noe, Nick Tansino

Overview:

For this project, we plan to implement a subset of UwUpp (also known as UwU++) in Python. UwU++ is an esoteric programming language based around unconventional, meme-related slang. As described by its creator, UwU++ aims to compete with Brainf*ck to be the most painful programming language to read and write in. The reason being that the entire programming language is written in what is known as UwU-speak, an English dialect that is phonetically spelled and pronounced as if spoken/written by an infant.

We plan to implement a few features that differentiate our subset from its predecessor. For example, while our implementation of UwU++ functions and conditionals will behave similarly to their predecessors, they will also include several distinctions involving behavior and structure. An example of this will be shown through our subset's ability to support functions as first-class citizens. We also plan to modify the configuration of UwU++ conditionals. Considering UwU++ doesn't seem to support the use of "else/else if" statements, our subset aims to support the use of these statements to provide users with alternative design options.

Our implementation design also involves several restrictions that differentiate our subset and UwU++, as well as its predecessor, UwU-lang. Regarding data types, our language will only support three primary data types (ints, floats, and strings). We also plan to revise UwU++'s I/O behavior by prohibiting alphabetic and/or alphanumeric input. Our subset will only deal with numerical user input and will raise an error otherwise.

Language Example:

Hello World:

```
UwU This is a basic hello world program
nuzzels("Hello World!")
```

In this simple program, nuzzels represents the typical print function.

Program 1:

```
UwU Program 1 from A5 in UwU++
x iws 1
nuzzels(x)
```

Assignment statements use iws (the phonetic pronunciation of 'is') as the '=' operator.

Conditional Program:

UwU Basic conditional program

```
nuzzels("Entew a vawue pwease: ")
x iws wisten()
*notices x*
    nuzzels(1)
*ewse*
    nuzzels(3)
```

stawp

* notices ... * represents a conditional statement. Stawp ends a conditional statement or loop.

While Loop Program:

UwU Simple program based on program_4.

UwU Demonstrates while-loop

```
nuzzels("Entew a vawue pwease: ")
x iws wisten()
OwO *notices 1 wess twan x *
    nuzzels(x)
    x iws x miwus 1
```

stawp

OwO represents a while loop. This program is based off of program 4 from assignment 5. It prints the values from x to 1 to the terminal.

Fibonacci:

UwU Fibonacci program

```
nyaa *fib(n)*
    UwU Declare the fib function
    *notices a gweatew twan 1*
        num iws fib(n minwus 1) pwus fib(n minus 2)
    stawp
    *notices n eqwall twoo 1*
        num iws 1
    stawp
    *notices n eqwall twoo 0*
```

```
num iws 0
stawp
wetuw n num
```

```
UwU Print the result of fib(5)
nuzzels(fib(5))
```

Nyaa ** declares a new function.

Milestone Schedule:

- UwU++ Lexer - December 6th, 2020
 - UwU++ Grammar - December 13th, 2020
 - UwU++ Parser - December 13th, 2020
 - UwU++ State - December 20th, 2020
 - UwU++ Interpreter - December 20th, 2020
-

Resources Required: In terms of hardware, we both require a computer capable of running a desktop operating system. Both of us need a keyboard, mouse, and headset. In terms of software, our team plans to use visual studio code as our IDE. Additionally, we will use Anaconda and Jupyter notebooks. Our group will implement UwU++ using Python3 with the Ply and Yacc libraries. Our team will also use the UwU++ GitHub repo as the source we base our implementation on.

Team Members & Tasks:

Team: Danny Noe and Nick Tansino

- UwU++ Lexer: (Danny & Nick)
 - Danny:
 - Function Definitions (def t_ID, def t_error, etc)
 - Building the lexer
 - Nick:
 - Reserved keywords
 - Tokens
 - Literals

- UwU++ Grammar: (Nick)
- UwU++ Parser: (Danny)
- UwU++ State: (Danny)
- UwU++ Interpreter: (Danny & Nick)
 - Danny:
 - Walk expression function
 - Variable scoping implementation
 - Conditional definitions
 - Nick:
 - Front-end / driver
 - Command Definitions (nuzzels, stawp, etc)
 - Dispatch Dictionary
 - Binary Operation Definitions