Easymoji Project Proposal

By Maxine Lui and Jocelyn Tseng

For 17-363 Programming Language Pragmatics Final Project

Project of Choice: Design a language construct and specify its semantics.

Problem: Finding emoji's when texting is tedious, and sometimes you want a preset list of emoji's to use to express your feelings, or spam several.

Solution: Easymoji is a web application that allows you to quickly type up long texts with plenty of emojis, and you can even save and reuse your favorite emoji combinations. Simply copy and paste the output to Messenger, Slack, or whatever platform you want to emote some more on.

Relation to Programming Languages: We will be designing our own language, Moji, and specifying its semantics. We will also implement a lexer to turn a program into tokens, a parser to turn the tokens into an AST, and a compiler to turn the AST into the output. And example of the process from input to output is as follows.

Example of the Entire Process (From Moji to Text):

Input	Output
let happyBirthday = 🥳🎂🎉 // this is a comment: text for jasmine => Text goes here. It may consist of words or emoticons [:)], references to the macros, &happyBirthday, or built in emojis such as :angry: or :eyes:!!! And #5 :meditate: lol => And you can have multiple inputs at once! #3 :party:	Text goes here. It may consist of words or emoticons ☺, references to the macros, 🥳🎂🎉, or built in emojis such as 😠 or 👀!!! And 🧘🧘🧘🧘🧘 lol And you can have multiple inputs at once! 🎉🎉🎉

Lexer Output (given original input):

```
LET
MACRO happyBirthday
ASSIGN
STR "🥳🎂🎉"
START
STR "Text"
STR "goes"
STR "here."
STR "It"
STR "may"
STR "consist"
STR "of"
STR "words"
STR "or"
STR "emoticons"
EMOTE ":)"
STR ","
STR "or"
STR "built"
STR "in"
STR "emojis"
STR "such"
STR "as"
EMOJI angry
STR "or"
EMOJI eyes
STR "!!!"
STR "reference"
STR "to"
STR "the"
STR "macros,"
MACRO happyBirthday
STR ","
STR "or"
STR "built"
STR "in"
STR "emojis"
STR "such"
STR "as"
EMOJI angry
STR "or"
EMOJI eyes
STR "!!!"
STR "And"
SPAM 5
EMOJI meditate
STR "lol"
START
STR "And"
STR "you"
STR "can"
STR "have"
STR "multiple"
STR "inputs"
STR "at"
STR "once!"
SPAM 3
EMOJI party
```

Parser Output (given Lexer's output as input):

```
{body =
    [(MacroDecl
       { macro_id = { id = {name = "happyBirthday"} };
         init = (Str "🥳🎂🎉")
       }
    );
    (TextDecl
       { words = [(Str "Text");
                  (Str "goes");
                  ...
                  (Str "emoticons");
                  (Emote ":)");
                  ...
                  (Macro { id = {name = "happyBirthday"} });
                  ...
                  (SpamDecl
                     { count = (Num 5.);
                       emoji = (Emoji {{ id = {name = "meditate"} }});
                     }
                  );
                 ]
       }
    );

    (TextDecl
       { words = [(Str "And");
                  (Str "you");
                  (Str "can");
                  (Str "have");
                  (Str "multiple");
                  (Str "inputs");
                  (Str "at");
                  (Str "once!");
                  (SpamDecl
                     { count = (num 3.);
                       emoji = (Emoji {{ id = {name = "party"} }});
                     }
                  );
                 ]
       }
    );

    ]
}

```

Compiler: takes parser's output as input and outputs the actual text or list of texts

Sketch of Concept for Web Interface: