Mini Basic v2.0

此项目在此前 Mini Basic 的基础之上，继续增加了三个新的需求：

1. 语法错误高亮

在点击 RUN 进行运行或者进入调试模式之后，应首先完成所有语句的语法树解析，并在“代码”显示框中，将有语法错误的代码行进行高亮（将对应行的底色设为红色）

HINT：TextEdit 中的高亮示例：

```
// 获取需要高亮的 TextEdit 对象（QTextBrowser 是 QTextEdit 的只读子类）
QTextBrowser *code =    ui->CodeDisplay;
QTextCursor cursor(code->document());

// 用于维护的所有高亮的链表
QList<QTextEdit::ExtraSelection> extras;
QList<QPair<int, QColor>> highlights = {
    {92, QColor(100, 255, 100)},
    {131, QColor(255, 100, 100)},
    {180, QColor(255, 100, 100)}
};

// 配置高亮，并加入到 extras 中
for (auto &line : highlights) {
    QTextEdit::ExtraSelection h;
    h.cursor = cursor;
    // 下面这些的功能，请大家自行查看文档
    h.cursor.setPosition(line.first);
    h.cursor.movePosition(QTextCursor::StartOfLine);
    h.cursor.movePosition(QTextCursor::EndOfLine);
    h.format.setProperty(QTextFormat::FullWidthSelection, true);
    h.format.setBackground(line.second);
    extras.append(h);
}

// 应用高亮效果
```

```
code->setExtraSelections(extras);
```

2. 支持字符串类型和格式化输出

2.1. 字符串类型变量定义

字符串类型的变量同样使用 LET 命令进行定义，其中的字符串应使用 “” 或者 ‘’ 包围。
例如:

```
LET s = "hello world"
LET t = 'Mini Basic'
```

BASIC 中的字符串中不能包含单引号和双引号（"和'），如果出现，则在执行时应报错。

2.2. 字符串类型的输入

为了接受用户的字符串输入，在原有命令的基础上，增加一个 INPUTS 命令， 其与 INPUT 命令的功能要求类似。不同的是 INPUT 用于让用户输入一个整型变量，INPUTS 则要求用户输入一个字符串类型的变量。

由于引入了新的变量类型，在解释器执行代码时，需进行简单的运行时类型检查，即字符串类型变量无法进行表达式运算。

2.3. 格式化输出

增加 PRINTF 命令，用于格式化输出。PRINTF 接受 1 到 N 个参数，其中第一个参数需为字符串类型，为格式字符串。格式字符串中可能会包含 0 个或 N-1 个 “{}” 作为占位符，依次对应此后的 N-1 个参数。占位符的个数应与此后参数个数严格对应，否则执行时报错。
为了实现的简单， 格式化字符串中“{}”只作为占位符出现，不存在单独的“{”或者“}”。

多个参数之间以一个英文逗号隔开。PRINTF 在打印的末尾会自动加换行。

一些示例如下：

```
PRINTF "Hello World"
```
输出：Hello World

```
PRINTF "Mini Basic V {}", 2
```
输出：Mini Basic V 2

```
PRINTF 'hello {}', 'world'
```
输出：hello world

```
PRINTF "{} {} {}", "hello", 2, "world"
```
输出：hello 2 world

```
LET a = 100
PRINTF "a = {}", a
```
输出：a = 100

3. 支持单步调试

3.1. 增加“调试/单步（Debug/Step)”按钮，实现单步调试的功能。

■ 在非调试模式下，点击 Debug 按钮后将进入调试模式（此时尚未执行任何一行 Basic 代码），高亮（底色为绿色）下一条将被执行的代码（即第一条被执行的代码）。

■ 在调试模式下，点击 Step 按钮，则进行“单步”操作，即执行一行代码，并更新代码高亮的显示。

在调试模式下，“语句与语法树”中应只显示当前（即下一次要执行的行）的语法树。

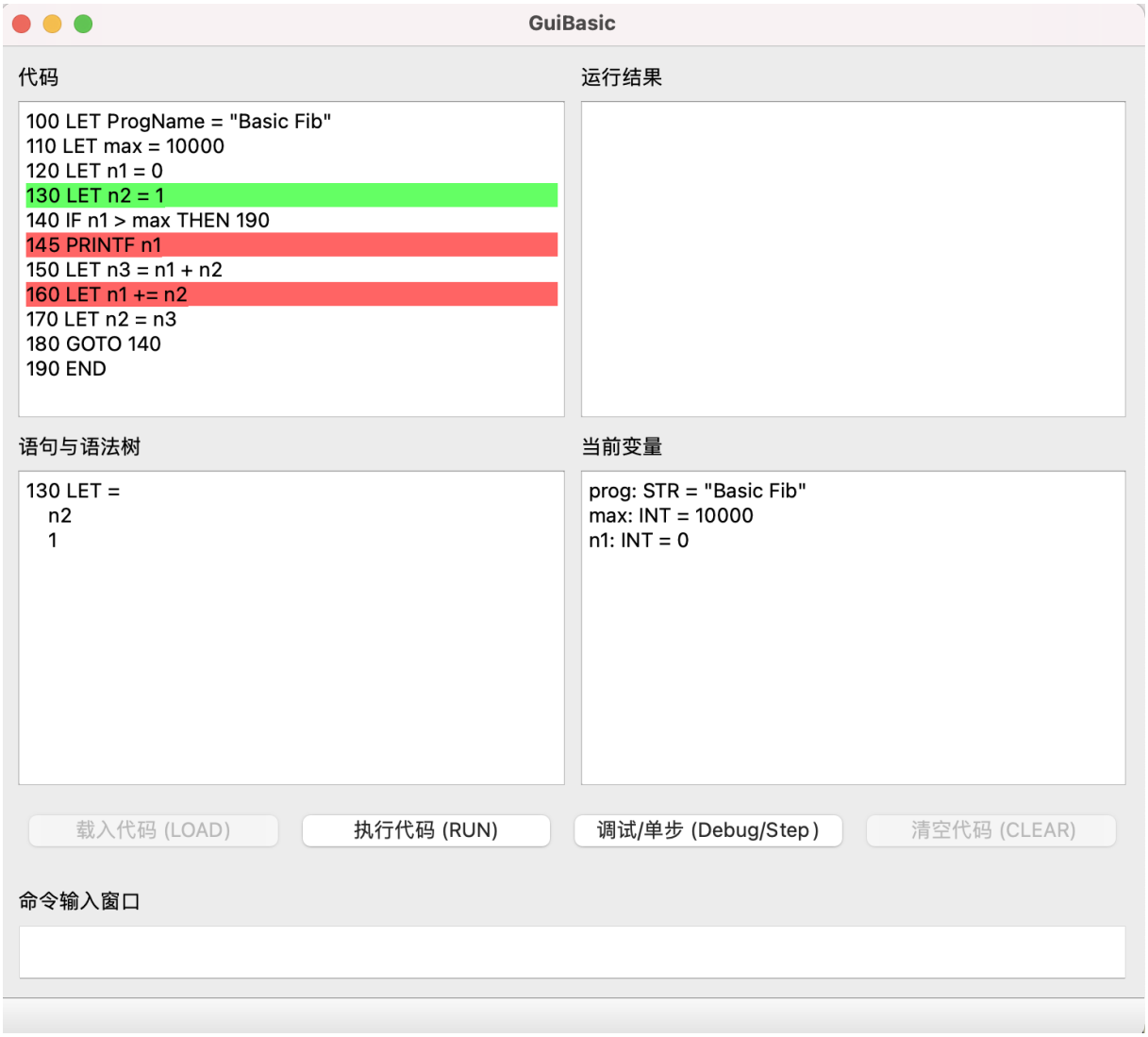
在调试模式下，“载入代码”和“清空代码”按钮将变为不可用（enabled 为 false），但“执行代码”按钮依然可用，效果是继续执行，直到 BASIC 程序运行结束。

当调试模式中执行到有错误的 BASIC 代码，或者程序结束后，应跳出弹窗，表示“该语句有错误”或“被调试的程序正常结束”，此后程序退出调试模式。

3.2. 增加“当前变量”显示部件。

不论是否在调试模式，该部件中始终显示当前 Context 中所有已经被定义的变量和其类型与值。即在正常 RUN 之后，该部件显示在运行结束后（不论正常还是异常结束）的变量情况。

一个程序运行的示例：



对于此文档中未规定到的部分，可以自行发挥，最终答辩时说明即可。