SuperPILOT IDE - User Guide

Version 1.0

SuperPILOT Development Team

Table of Contents

1. Introduction

What is SuperPILOT IDE?

SuperPILOT IDE is a comprehensive development environment that combines three classic educational programming languages in one powerful tool:

- **PILOT**: Programmed Inquiry, Learning, or Teaching language
- **Logo**: Educational programming language with turtle graphics

- **Tiny BASIC**: Simplified version of the BASIC programming language

This IDE provides a modern interface for learning and experimenting with these classic languages while offering advanced features like debugging, project management, and integrated turtle graphics.

Who Should Use SuperPILOT IDE?

SuperPILOT IDE is designed for:

- Students learning programming fundamentals
- Educators teaching computer science concepts
- Retro computing enthusiasts
- Anyone interested in classic programming languages
- Developers exploring educational programming paradigms

Key Features

- Multi-language support in one environment
- Integrated turtle graphics with visualization
- Professional debugger with breakpoints
- Project management system
- Syntax highlighting and line numbering
- Comprehensive help system with examples
- Export capabilities to Python
- Keyboard shortcuts for efficient workflow

2. Installation

System Requirements

SuperPILOT IDE requires the following system specifications:

- Operating System: Debian Trixie or compatible Linux distribution
- Processor: 1 GHz or faster
- Memory: 512 MB RAM minimum
- Storage: 100 MB available space

- Python: Version 3.8 or higher

Required Dependencies

The following Python packages are required:

- `python3-tk` : Tkinter GUI toolkit
- `python3-pil` : Python Imaging Library
- `python3-pil.imagetk` : PIL support for Tkinter

Installation Steps

Step 1: Install Required Packages

Open a terminal and run the following commands:

```
sudo apt update
sudo apt install python3 python3-tk python3-pil python3-pil.imagetk
```

Step 2: Download SuperPILOT IDE

Download the `superpilot.py` file to your preferred directory.

Step 3: Make Executable

Make the file executable:

```
chmod +x superpilot.py
```

Step 4: Run the Application

Execute the application:

```
./superpilot.py
```

3. IDE Interface

Main Window Components

Menu Bar

The menu bar provides access to all IDE features:

- **File**: New, Open, Save, Save As, Recent Files, Exit
- **Edit**: Undo, Redo, Cut, Copy, Paste, Find, Replace
- **Run**: Run, Run with Debugger, Stop, Step controls
- **Project**: New Project, Open Project, Project Settings
- **Tools**: Turtle Graphics, Clear Output, Reset Interpreter, Syntax Check
- **Examples**: PILOT, Logo, and BASIC examples
- **Help**: Documentation and support

Toolbar

The toolbar provides quick access to common operations:

- New, Open, Save buttons
- Run, Stop, Debug buttons
- Turtle Graphics, Clear Output buttons

Editor Tab

The editor tab features:

- Line-numbered text editor
- Full text editing capabilities
- Scrollbars for navigation

Output Tab

The output tab displays:

- Program output text
- Error messages
- Debug information
- Execution results

4. PILOT Language Reference

Overview

PILOT (Programmed Inquiry, Learning, or Teaching) is a language designed for computer-assisted instruction. It uses a simple command structure with single-letter prefixes.

Basic Commands

Command	Description	Example
T:text	Output text	T:Welcome to PILOT!
A:variable	Accept input	A:NAME
Y:condition	Match if true	Y:*SCORE* > 80
N:condition	Match if false	N:*NAME* == ""
J:label	Jump to label	J:SUCCESS
M:label	Jump if match flag set	M:CORRECT
R:label	Gosub to label	R:SUBROUTINE
C:	Return from subroutine	C:
L:label	Label definition	L:START
U:var=expr	Update variable	U:SCORE=*SCORE*+10
END	End program	END

Variables

In PILOT, variables are referenced using asterisks:

- Variable names are case-sensitive
- Use `*VARNAME*` format when referencing in text
- Example: `T:Hello *NAME*, your score is *SCORE*`

Example Program

```
L:START
T:Welcome to the Math Quiz!
A:NAME
T:Hello *NAME*! Let's begin.
U:SCORE=0
U:QUESTION=1

L:QUESTION1
T:Question 1: What is 2+2?
A:ANSWER
Y:*ANSWER* == 4
J:CORRECT1
N:*ANSWER* != 4
J:WRONG1

L:CORRECT1
T:Correct! +10 points
U:SCORE=*SCORE*+10
U:QUESTION=*QUESTION*+1
J:QUESTION2

L:WRONG1
T:Sorry, that's incorrect.
U:QUESTION=*QUESTION*+1
J:QUESTION2

L:QUESTION2
T:Question 2: What is 5*3?
A:ANSWER
Y:*ANSWER* == 15
J:CORRECT2
N:*ANSWER* != 15
J:WRONG2

L:CORRECT2
T:Correct! +10 points
U:SCORE=*SCORE*+10
J:RESULTS

L:WRONG2
T:Sorry, that's incorrect.
```

```
J:RESULTS

L:RESULTS
T:*NAME*, your final score is *SCORE*.
Y:*SCORE* >= 20
T:Excellent work!
N:*SCORE* < 20
T:Keep practicing!
END
```

5. Logo Language Reference

Overview

Logo is a programming language designed for educational use with turtle graphics.

Basic Movement Commands

Command	Abbreviation	Description	Example
FORWARD n	FD n	Move forward n steps	FORWARD 100
BACK n	BK n	Move backward n steps	BACK 50
LEFT n	LT n	Turn left n degrees	LEFT 90
RIGHT n	RT n	Turn right n degrees	RIGHT 45
PENUP	PU	Lift pen up	PENUP
PENDOWN	PD	Put pen down	PENDOWN
CLEARSCREEN	CS	Clear drawing	CLEARSCREEN
HOME		Move to center	HOME

Position and Heading

Command	Description	Example
SETXY x y	Move to position	SETXY 100 50
SETHEADING n	Set heading	SETHEADING 0
SETH n	Set heading	SETH 90

Example Program

```
PENCOLOR "blue"
FD 100
RT 90
FD 100
RT 90
FD 100
RT 90
FD 100

PENUP
FD 200
PENDOWN
PENCOLOR "red"

TO TRIANGLE SIZE
  REPEAT 3 [FD SIZE RT 120]
END

TRIANGLE 100
```

6. Tiny BASIC Language Reference

Overview

Tiny BASIC is a simplified version of the BASIC programming language.

Basic Commands

Command	Description	Example

LET var = expr	Assign value	LET X = 10
PRINT expr	Output value	PRINT "Hello"
INPUT var	Get input	INPUT NAME
GOTO line	Jump to line	GOTO 100
IF condition THEN cmd	Conditional	IF X>10 THEN PRINT "BIG"
FOR var=start TO end	Loop start	FOR I=1 TO 10
NEXT var	Loop end	NEXT I
GOSUB line	Subroutine call	GOSUB 200
RETURN	Subroutine return	RETURN
END	End program	END
REM comment	Comment	REM This is a comment

Example Program

```
10 PRINT "NUMBER GUESSING GAME"
20 PRINT "I'M THINKING OF A NUMBER BETWEEN 1 AND 100"
30 LET SECRET = INT(RND(1)*100)+1
40 LET GUESSES = 0
50 GUESSES = GUESSES + 1
60 INPUT "YOUR GUESS: "; GUESS
70 IF GUESS = SECRET THEN GOTO 120
80 IF GUESS < SECRET THEN PRINT "TOO LOW!"
90 IF GUESS > SECRET THEN PRINT "TOO HIGH!"
100 GOTO 50
110 REM GAME LOOP
120 PRINT "CORRECT! YOU GUESSED IN "; GUESSES; " TRIES"
130 IF GUESSES < 5 THEN PRINT "EXCELLENT!"
140 IF GUESSES >= 5 AND GUESSES < 10 THEN PRINT "GOOD JOB!"
150 IF GUESSES >= 10 THEN PRINT "KEEP PRACTICING!"
160 END
```

7. Debugger

Debugger Features

- Breakpoint management
- Step execution (Over, Into, Out)
- Variable inspection
- Call stack view
- Real-time execution monitoring

Debugger Controls

Key	Function
F5	Run program normally
F6	Run with debugger
F8	Stop execution
F10	Step over
F11	Step into
Shift+F11	Step out

8. Examples

PILOT Example: Simple Quiz

```
L:START
T:Welcome to the Animal Quiz!
A:NAME
T:Hello *NAME*!

L:QUESTION1
T:What sound does a cow make?
```

```
A:ANSWER
U:ANSWER=*ANSWER*.lower()
Y:*ANSWER* == "moo"
J:CORRECT1
N:*ANSWER* != "moo"
J:WRONG1

L:CORRECT1
T:Correct! A cow says moo.
J:ENDQUIZ

L:WRONG1
T:Incorrect. A cow says moo.
J:ENDQUIZ

L:ENDQUIZ
T:Thanks for playing, *NAME*!
END
```

Logo Example: Drawing Patterns

```
TO SPIRAL SIZE ANGLE
  MAKE "DISTANCE 1
  REPEAT 100 [
    FD :DISTANCE
    RT :ANGLE
    MAKE "DISTANCE :DISTANCE + SIZE
  ]
END

PENCOLOR "purple"
SPIRAL 2 91

PENUP
SETXY 200 0
PENDOWN
PENCOLOR "green"
SPIRAL 1 89
```

BASIC Example: Number Guessing Game

```
10 PRINT "NUMBER GUESSING GAME"
20 PRINT "I'M THINKING OF A NUMBER BETWEEN 1 AND 100"
30 LET SECRET = INT(RND(1)*100)+1
40 LET GUESSES = 0
```

```
50 GUESSES = GUESSES + 1
60 INPUT "YOUR GUESS: "; GUESS
70 IF GUESS = SECRET THEN GOTO 120
80 IF GUESS < SECRET THEN PRINT "TOO LOW!"
90 IF GUESS > SECRET THEN PRINT "TOO HIGH!"
100 GOTO 50
110 REM GAME LOOP
120 PRINT "CORRECT! YOU GUESSED IN "; GUESSES; " TRIES"
130 IF GUESSES < 5 THEN PRINT "EXCELLENT!"
140 IF GUESSES >= 5 AND GUESSES < 10 THEN PRINT "GOOD JOB!"
150 IF GUESSES >= 10 THEN PRINT "KEEP PRACTICING!"
160 END
```

9. Troubleshooting

Common Issues

IDE Won't Start

> **Problem:** Missing dependencies
> **Solution:** Install required packages:
>
> ```
> sudo apt install python3-tk python3-pil python3-pil.imagetk
> ```

Turtle Graphics Not Appearing

> **Problem:** Turtle window hidden
> **Solution:** Go to **Tools** → **Turtle Graphics** or include turtle commands in your program

Programs Running Too Fast

> **Problem:** Infinite loops or long execution
> **Solution:** Use debugger (F6) or add breakpoints

Error Messages

"Syntax Error"

- Check for missing colons in PILOT commands
- Verify proper line numbering in BASIC
- Ensure balanced quotes and parentheses

"Variable Not Defined"

- Check variable spelling and case
- Ensure variables are initialized before use

"Label Not Found"

- Verify label spelling matches exactly
- Ensure labels are defined with proper syntax

Keyboard Shortcuts

Shortcut	Function
Ctrl+N	New file
Ctrl+O	Open file
Ctrl+S	Save file
Ctrl+Shift+S	Save as
Ctrl+Z	Undo
Ctrl+Y	Redo
Ctrl+F	Find
Ctrl+H	Replace
F5	Run program

F6	Run with debugger
F8	Stop execution
F10	Step over
F11	Step into
Shift+F11	Step out

For support, contact: support@superpilot.org