GameMaker BASIC Interpreter – TODO List

* ✔ Implement expression parsing for IF ... THEN statements using `basic\_evaluate\_expression`.
* ✔ Fix PRINT to skip parsing anything inside double quotes.
* ✔ Fix LET to support functions like RND(), ABS(), and math expressions.
* ✔ Add command `BGCOLOR` to set background color using named or RGB values.
* ✔ Update DRAW event in obj\_basic\_interpreter to honor `global.background\_draw\_enabled` and `global.background\_draw\_color`.
* ✔ Maintain a global color map (`global.colors`) shared by both COLOR and BGCOLOR commands.
* ✔ Ensure background drawing works when color is set, and doesn’t interfere with text output.
* ☐ Add a full expression parser to eventually replace token-based math logic.
* ✔Add cleanup at program end: destroy all stacks, maps, and lists cleanly.
* ✔ Add optional `BGCLEAR OFF` command to disable background fill when needed.
* ☐ Evaluate future support for background gradients or images in a future version.
* **✅ GameMaker BASIC Interpreter — Master TODO List**
* **🟢 Implemented Core Features**
* These are already working as of now:
* PRINT (with + concatenation and ; to suppress newline)
* LET (basic variable assignment, some math expressions)
* INPUT
* COLOR
* CLS
* GOTO
* GOSUB / RETURN
* FOR / NEXT
* WHILE / WEND
* END
* MOD, ABS, and RND (via basic expression parser)
* **🔧 CURRENT + PLANNED TODOs**
* **🧠 Expression Parsing (Major Rework Needed)**
* **Implement a full expression parser**  
  Current evaluation only supports very simple expressions like A + B or RND(10). We need support for:
  + Nested parentheses: LET X = (RND(10) + 1) \* 2
  + Multiple operators: LET A = B + C \* 3
  + Operator precedence and grouping
* Evaluate expressions in **LET**, **IF**, and **PRINT** consistently
* **~~🧼 Runtime Cleanup & Safety~~**
* **~~Destroy all ds\_\* structures~~** ~~(stacks, maps, lists) when the interpreter exits~~
  + ~~global.basic\_variables~~
  + ~~global.output\_lines, global.output\_colors~~
  + ~~gosub\_stack, while\_stack, for\_stack~~
* ~~Hook this into room end or reset logic~~
* **🧾 Enhanced Command Support**
* ~~REM — Support comment lines that do nothing but allow in program listing~~
* RND improvements: Seedable RNG, better float control
* Add INT() to truncate decimal values
* Add LEN(), LEFT$(), RIGHT$(), MID$() for string handling
* Add STR$() and VAL() for type conversion
* Add INKEY$ or equivalent for single character input
* **🧑‍💻 Editor Features & Usability**
* **Improved error messages** (include line number and better context)
* Add syntax highlighting or command hinting in the editor
* Add a SAVE / LOAD feature for storing BASIC programs locally
* Allow export to .bas file format
* Add scrollable output panel and better history management
* **🎨 Aesthetic Enhancements**
* Add BEEP or sound effects
* Add font size/color customization for PRINT
* Add theme toggles (retro green, amber, etc.)
* **🧪 Testing & Demos**
* Build comprehensive demo programs using all commands
* Create stress tests for nested loops and subroutines
* Add example programs to paste via :PASTE
* **☁️ GitHub / Source Control Workflow**
* Write proper .gitignore for GMS projects:
* lua
* CopyEdit
* \*.yy~
* \*.yyp~
* .\*.backup
* .gml.cache/
* .gitattributes
* config/
* output/
* Automate README updates with new commands
* Add screenshots of interpreter + sample program
* **🧱 Architectural Improvements**
* Refactor basic\_cmd\_let() to delegate to basic\_evaluate\_expression()  
  (Remove duplicate evaluation logic from LET, IF, PRINT)
* Unify command dispatch system (handle\_basic\_command) using a cleaner map/registry approach
* **🔍 Open Questions**
* Should the RND function return float or int by default? (Current: int only)
* Should we support DO/LOOP or REPEAT/UNTIL constructs in future?