Overview

Game Engine is an intricate program that merely creates a string that can represent a game, (not limited). Nonetheless, the string doesn't instantly create a game when pasted into a program file, it takes programming and skills to actually make it work.

Data Storage Structure

My Program (Game Engine) Has a 3 matrix structure. This means that it stores data in 3 matrixes.

1. Properties:
   - -User Control
   - -Is Barrier?
   - -Kills Player?
   - -Movement (after decimal point)
2. Point IDs

   Every Point in the matrix (game) should have a 2 digit point ID. When Appending Properties or movement, all Points with the selected point ID will receive the same properties.

3. Colors

   Colors will be set in the same manner as the properties, based on the selected point ID.

Compiling

To compile, just hit 'graph' the program will take somewhere around a minute to compile 1200 instances worth of properties to a single string.

Decompiling

Square matrix is supposed to save you the time of hard coding a set of occurrences into a program, this is essentially why you aren't limited to just making games with this program, an experienced programmer may want to use the program to create a spreadsheet, or maybe some other useful program. Therefore, I have designed a few useful things into the program.

PROPERTY CORRELATION.

There are 3 basic properties in my program. However, one may see that you cant have a user-controlled point, being a barrier, or something killing the player and not being a barrier (barriers can technically move; see movement implementation).

So you have already read that there are 3 main properties and a movement property. The property matrix stores your properties as a 3 digit number with 1 decimal point.

For example: 121.0 would represent a point with the following property values:

User Control	False
Is Barrier	True
Kills Player	False
(Movement)	None

However: 211.0 is a bit more complex you have a user controlled point with a 0 in the fractional part signifying no movement. This is the beauty of Square Matrix/ Game Engine. It is up to the user to interpret the properties, not the program.

However, if you are going to use the property values of the default program intentions. Here are some tips to understand how it works:

1. User Control cannot have the same toggle with any of the other properties.
2. If Kills Player is true, then Is Barrier is also true.

These instances are always in play. Even if you aren't planning on making a game, and want the properties to mean something else.

DECOMPILING AUTOMATION

If you don't want to make a custom use for the matrix you are creating, that's OK. However, if you are trying to make a game with this tool here is some useful information:

My program doesn't ruin the fun of actually getting to make a program, for any use. However, if you are trying to use the basic program properties decrypting the *very* long hex code string file.

This is why my program has a decompilation routine, all that a developer must do to decrypt the string into the original 3 matrixes is enter the following code and then call my program:

```
:(π^3)*256.23232323–>θ
:prgmSQARgame
```

This will automatically start the Square Matrix program's decrypting logic, which will successfully decrypt the string into the 3 original matrixes, however, it is advised that you do not use this logic, as it takes time and requires the presence of the SQARgame program on the users' calculator, therefore, I advise that you manually decrypt the program in your own interpretation of the hex code string.

Output String Format

The Output string can be normally decoded as follows *(basing this off of the regular properties, movements, and color):*

Example:
...78TFF1A...

Point ID	78
Properties in Order	TFF
Movement	1
Color	A (blue)