ProGame 0 ReadMe

_____

_____

========
Contents
========

1. Prereqs
2. The program; How it works
3. How to play this game (and win)
4. Credits

_____

_____

==========
1. Prereqs
==========

        This program requires nothing special, just a TI-84 Plus CE (it might
work on the +CSE too; I haven't tested it on one), 8+ KB of RAM, and the TI
Connect CE software (https://education.ti.com/en/us/products/computer_software/
connectivity-software/ti-connect-ce-software/tabs/overview).

It takes just 4 steps to put it on your calculator:

1. Extract this .zip folder if you haven't already, and open the folder that
your computer extracted it to.
2. Open TI Connect CE, plug your calculator into your computer, and click
on the "CALCULATOR EXPLORER" tab on the side.
3. Drag the "PROGAME0.8xp" file into the list of files in the TI Connect window,
as well as the three .8xl files (PROCD.8xl, PROGP.8xl, and PROGT.8xl).
4. Click SEND without changing the settings that the pop-up dialogue box gives you (but make
sure the destination is RAM, not Archive!).

If you did it right, the program and 3 new lists should now be on the calculator. If
my instructions don't help you, ask someone else.

_____

_____

===========================
2. The program; How it works
===========================

        This program is a BASIC game intended for a short period of "fun and
games". It is not meant to be "the most amazing calculator game ever" (mostly
because people who say that their calculator games are the best usually make
games that are a load of garbage), nor do I think it ever will be. The play
time it takes is approximately 3-10 minutes, depending on your skill level
(and also how much of this ReadMe you actually read): However, do not force
beating the game in one sitting upon yourself; as soon as one level is over,
your progress is saved to one of the custom lists.
        The program works based on data you insert into the program, as well
as random number generation (RNG). The RNG used in the program reads the data
you inserted and "interprets" it, leaving some data intact while other data
is skewered. This "interpreted" data is now read as movement information,
which changes some variables to reflect this movement. The whole test is

given a "game" element with a "target" that these movements must reach.
        There are 5 save profiles available to the user of the program (not
because the user will need them, but "because."), all of which are
retained in one of the custom lists. They do not completely save your
progress, but they save which level you are on so you can go back and
restart that level next time you load the profile.

_____

_____

================================
3. How to play this game (and win)
================================

============
SPOILER ALERT
============

==
The next part of this ReadMe includes code/pseudo-code and other information
which may ruin the random element of the game. If you don't want to know,
don't continue.
==

The first level has code arranged thus -

Code:

```
If |INPUTCODE(1)>2:randInt(2,3)→|INPUTCODE(1
If |INPUTCODE(2)>1:randInt(1,3)→|INPUTCODE(2
If |INPUTCODE(3)>2:randInt(0,3)→|INPUTCODE(3
If |INPUTCODE(4)>3:randInt(0,1)→|INPUTCODE(4
....
If |INPUTCODE(x)=0:row-1→row
If row<3:8→row
If |INPUTCODE(x)=1:row+1→row
If row>8:3→row
If |INPUTCODE(x)=2:column-1→column
If column<3:8→column
If |INPUTCODE(x)=3:column+1→column
If column>8:3→column
```

("|INPUTCODE" is the list of inputs the user types in, all of which are stored
to the same list. "x" is an ascending incremental value going from 1 to 4)

Analysis (split into a tiered list):

"SERIES 1,1"
        |
        Guaranteed:
        You can go up (1), down (0), or left (2).

|
        Random:
        You can go left or right if you enter a value greater than 2.
"SERIES 1,2"
           |
        Guaranteed:
        You can go up (1) or down (0).
           |
        Random:
        You can go up, left, or right if you enter a value greater than 1.
"SERIES 1,3"
           |
        Guaranteed:
        You can go up (1), down (0), or left (2).
           |
        Random:
        You can go any direction if you enter a value greater than 2.

The second level has code arranged thus -

Code:

If ⌊INPUTCODE(1)≥2:randInt(2,3)→⌊INPUTCODE(1
If ⌊INPUTCODE(2)≤1:randInt(1,3)→⌊INPUTCODE(2
If ⌊INPUTCODE(3)≥1:randInt(0,3)→⌊INPUTCODE(3
If ⌊INPUTCODE(4)<2:randInt(0,1)→⌊INPUTCODE(4
....
Movement code is universal; see above.

Analysis:

        This code is a little more tricky to manipulate. The following analysis
shows what you can try to do.

"SERIES 2,1"
           |
        Guaranteed:
        You can go up (1) or down (0).
           |
        Random:
        You can go left/right if you enter a value of 2 or higher.

"SERIES 2,2"
           |
        Guaranteed:
        You can go left (2) or right (3).
           |
        Random:
        You can go up, left, or right if you enter a value of 1 or lower.
"SERIES 2,3"
           |
        Guaranteed:
        You can go down (0).
           |
        Random:
        You can go all directions if you enter a value of 1 or greater.
"SERIES 2,4"
           |
        Guaranteed:

```
        You can go left (2) or right (3).
        |
        Random:
        You can go up/down if you enter a value less than 2.

-------------

The third level has code arranged thus -

Code:

If |INPUTCODE(1)≥2:randInt(1,3)→|INPUTCODE(1
If |INPUTCODE(2)≤1:randInt(1,2)→|INPUTCODE(2
If |INPUTCODE(3)<3:randInt(0,2)→|INPUTCODE(3
....

(only 3 inputs are accepted in this level)

Analysis:

"SERIES 3,1"
        |
        Guaranteed:
        You can go up (1) or down (0).
        |
        Random:
        You can go up, left, or right if you enter a value of 2 or greater.
"SERIES 3,2"
        |
        Guaranteed:
        You can go left (2) or right (3).
        |
        Random:
        You can go up or left if you enter a value of 1 or less.
"SERIES 3,3"
        |
        Guaranteed:
        You can go right (3).
        |
        Random:
        You can go up, down, or left if you enter a value less than 3.

-------------

The fourth (and final) level has code arranged thus -

If |PROCD(1)≥2:randInt(1,3)→|PROCD(1
If |PROCD(2)≤1:randInt(1,2)→|PROCD(2
If |PROCD(3)<3:randInt(0,2)→|PROCD(3
If |PROCD(4)=2:randInt(0,1)→|PROCD(4
If |PROCD(5)≠1:randInt(1,3)→|PROCD(5
....

Analysis:

"SERIES 4,1"
        |
        Guaranteed:
        You can go up (1) or down (0).
        |
        Random:
```

You can go up, left, or right if you enter a value of 2 or greater.
"SERIES 4,2"
        |
        Guaranteed:
        You can go left (2) or right (3).
        |
        Random:
        You can go up or left if you enter a value of 1 or less.
"SERIES 4,3"
        |
        Guaranteed:
        You can go right (3).
        |
        Random:
        You can go up, down, or left if you enter a value less than 3.
"SERIES 4,4"
        |
        Guaranteed:
        You can go up (1), down (0), or left.
        |
        Random:
        You can go up or down if you enter a value equal to 2.
"SERIES 4,5"
        |
        Guaranteed:
        You can go up (1).
        |
        Random:
        You can go up, left, or right if you enter a value other than 1.

_____

_____

==========
4. Credits
==========

        This code is © SMWAgent09AF May 2016. The "source code" may be distributed freely, but do not take credit for it without direct explicit permission from the author, or distribute it for gain or profit.

Programmer:
SMWAgent09AF

Code design:
SMWAgent09AF

DCSE icon:
SMWAgent09AF

Playtesting:
SMWAgent09AF,BobbyMcR

Troubleshooting:
SMWAgent09AF

Coded on TI-84 Plus CE v5.1.0.0110 and TI Connect CE v5.1.1.92

README generated on June 10, 2016

Thanks for downloading!
I'm on SMW Central if you need to contact me (my profile name is, of course,
SMWAgent09AF). I hope to make something a little more game like for TI 84 color
calculators soon. No promises, however.