HL_game_decomposition

Personal Private AO Invite

Intoduction

Display instructions for game + rules

Display DVLPR notes

Input

Ask for lower boundary number (VC)

Ask for higher boundary number (VC)

Number of rounds (individual HLGI's generated)

(InpSrt) Input for guess (first and beyond)

Generate the HLGI

Use random libraries to generate a number between higher and lower boundary.

Use logarithmic equation to count calculate minimal number of guesses required to get to the HLGI, number to be rounded up if resulting value is a decimal - set variable name to minguessHLGI

Calculate number of digits in the HLGI - set variable name to digHLGI

Generate number of guesses allowed - set variable name to guesallo: "minguessHLGI + (2 *digHLGI) = guesallo"

Compare user guesses to HLGI

If lower than high boundary >>

. If lower than lower boundary >>

.. Display "number out of given range of values" error message

.. Go to InpSrt

. Elif higher than lower boundary >>

.. If lower than HLGI >>

... Display "too low" message

... Go to InpSrt

.. Elif (higher than HLGI and lower than higher boundary) >>

... Display "too high" message

... Go to InpSrt

.. Else (guess input = HLGI) >>

... Display "congratulations" message

... End game (relevant statistics showcase)

Front-end Usability Mechanics 1

Record every guess made

Count number of guesses made - set variable name to guesmade

Calculate number of guesses left - set variable name to guesleft: "guesallo - guesmade = guesleft"

Report relevant statistics to user at the end of each guess made

Look into how to link a database to python

Front-end Usability Mechanics 2

User visual appeal

(Optional) Look into how to import images to python output

Notes

VC = Validity Check

HLGI = Higher/Lower Game Integer

InpSrt = Input Start (where the user is asked for input)

DVLPR = Developer

LPBG = Loop Beginning

LPBG2 = Loop Beginning 2

Extra: User Suggestion sub-program

Input for whether they have any suggestions to make or not >>

. If input = y, yes, yep, yup, yea or yeah >>

(LPBG) . Ask for suggestion input >>

.. If input >= (more than or equal to) 10 characters >>

(LPBG2) ... Say "Thank you for your suggestion!"

... Ask if they wish to add another suggestion

.... If input = y, yes, yep, yup, yea or yeah >>

..... Go to LPBG2

.... Elif input = n, no, nope, nah or na >>

..... Exit game and thank them for playing

.... Else say "Sorry I didn't get that. Please choose between Yes or No: "

.... Go to LPBG2

.. If input <10 characters (or just use an Else function) >>

... Say "Your suggestion needs to be at least 10 characters long. I appreciate your co-operation :) "

... Go to LPBG

. Elif input = n, no, nope, nah or na >>

.. Exit game and thank them for playing