



Help Emilia Gan with CS Research

QUESTIONS

RESPONSES

20

20 responses



SUMMARY

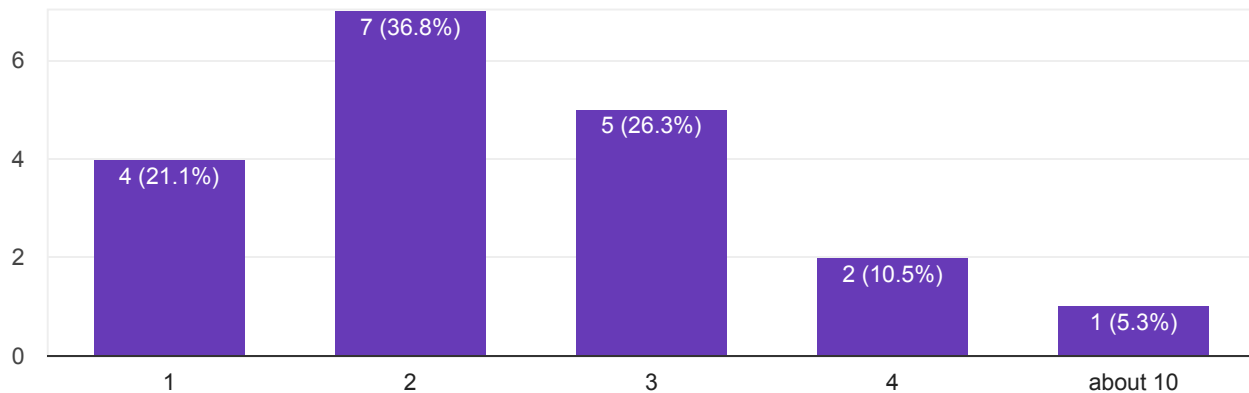
INDIVIDUAL

Accepting responses



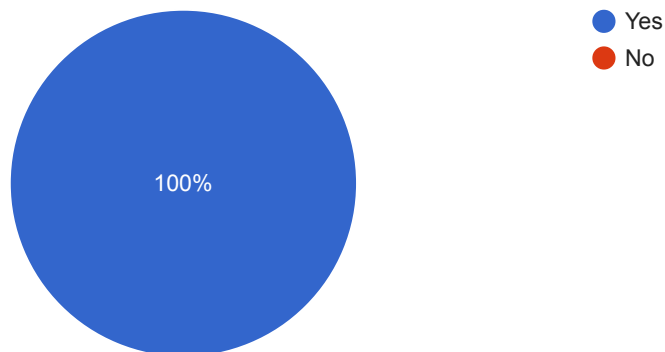
How many times did you play the game?

19 responses



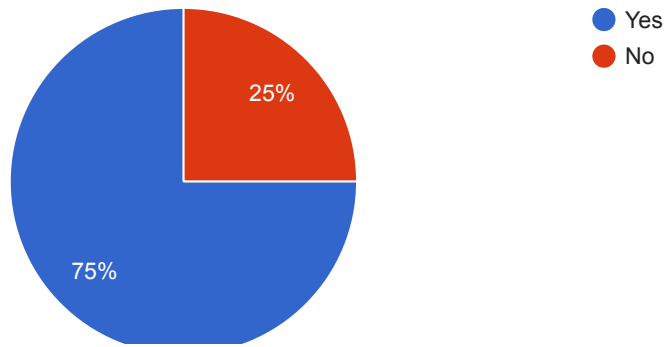
Were you already familiar with the idea of binary numbers?

20 responses



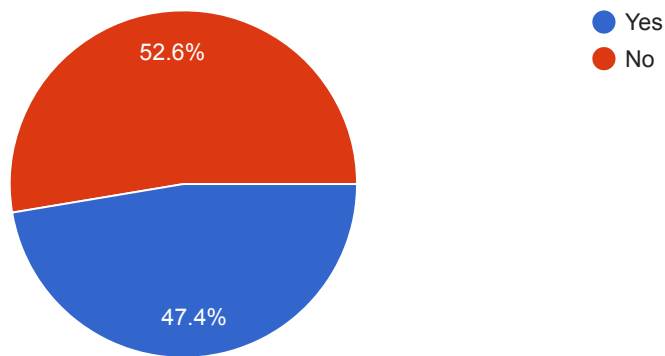
If you didn't know binary numbers, do you think you understand them now?

12 responses



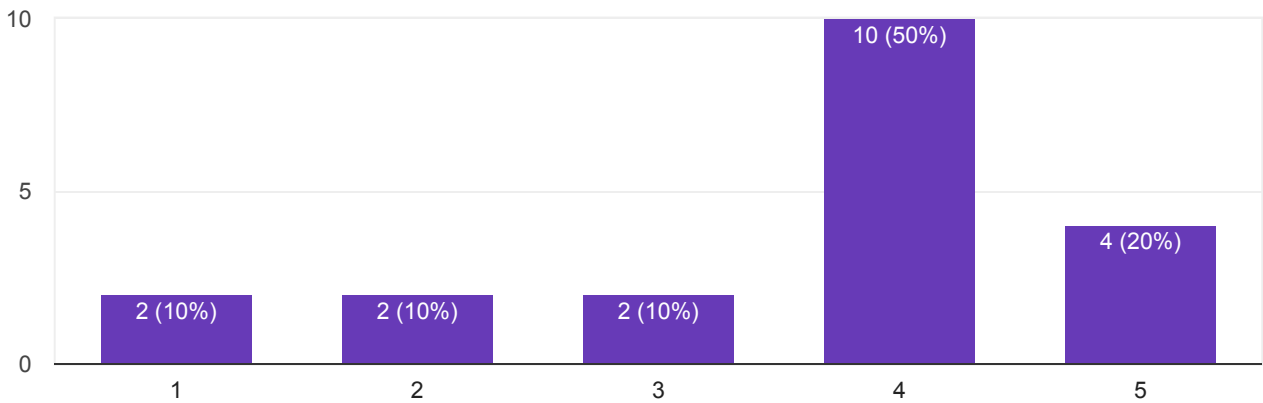
If you did know binary numbers, do you think you are more skilled at counting in binary now?

19 responses



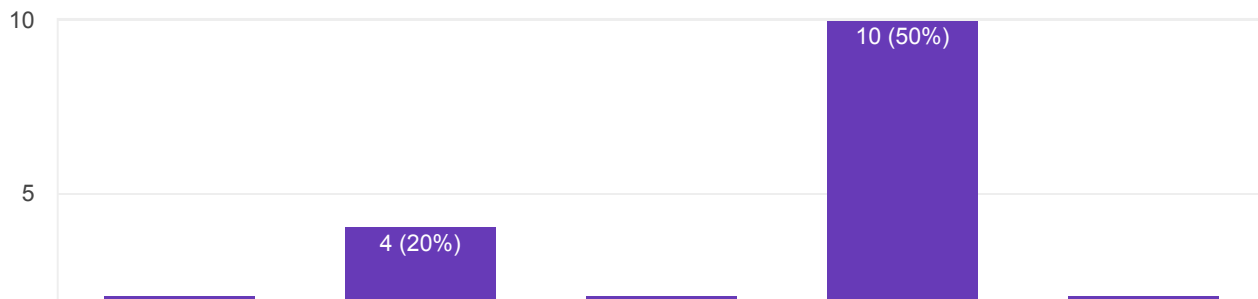
On a scale of 1 to 5, how educational would you say the game is?

20 responses



On a scale of 1 to 5, how entertaining would you say the game is?

20 responses



What suggestions would you have to improve the game?

17 responses

More sounds, and maybe multiplayer?

Popup is not a great way to get input

I don't know for what reason it didn't get harder, i.e. build chains of three or more

Make the directions more clear. Maybe allow to decide how long you want a play session to be. Or record every try so I can get on a roll.

Maybe something that tells you what the number is when you guess wrong. Or maybe a rule where you can't just guess the same number multiple times (I kept guessing 11111111 and 10000000)

Directions for those who are not familiar with binary ahead of time

A tutorial in the beginning for not only binary but also effective ways to play the game (pro tips). It's always a bad sign when you need to read a whole paragraph of text just to figure out how to play the game without actually being in the game - an experiential tutorial is far more effective.

Improve touch controls (it is very hard to choose the right numbers, and you can't redo your selection). I found that by continually choosing long 1-only strings I could very easily get to the Genius level. Not sure how that could be fixed though.

Instead of letting the user choose what the number they should guess, maybe give it to them and ask them what number it is with more time.

Selecting an arbitrary chain may result in getting easy scores. Rather than getting a large number, players should be given more points not only based on the resulting selected values. For example, selecting a chain of 1's make it easy for the players to "estimate" the numbers.

very bad game, hard to understand, then after that you jsut always pick 11, wouldnt let me select longer chains, also the popup boxes with the message were super annoying. overall pretty poorly designed game, i have no idea how this would help you learn honestly

Not sure how you would improve this, but I just looked for chains of 0 so I only needed to count the number of 0s and raise 2 to the power of that to get a "decent" score, which some people might do

none, really

Make it clear you have to drag the cursor, I was clicking each number.

Randomly suggest binary numbers instead of choosing them

I could cheat and get huge scores by selecting 1 as the first bit and following it with as many 0s I could find, and easy-mode my way by simply adding powers of two

force them to find a decimal number in the binary digits, otherwise you can just pick a sequence of 1's

What is the most enjoyable feature of the game?

12 responses

The fact that it gives you multiple chances. Thanks for that.

Looks cool, and works just fine on mobile

I like that I was able to choose and make it a just right challenge.

It's... fun? *shrug*

I can see how this might better enable you to count binary. I would've definitely preferred that the dragging of the numbers starts from the ones column - I.E. dragging across 1 - > 1 -> 0 gives you the binary number 011. That way, you can add as you go whereas the current configuration means you have to recalculate what each column is worth instead of being able to do it in your head.

The "levelling system"

Ranks of points

trying to choose a path that's easy to count!

Finding out that I could drag over more than 4 numbers at a time.

Actually calculating the decimal

finding the best way to select as many 0s as possible

boggle like