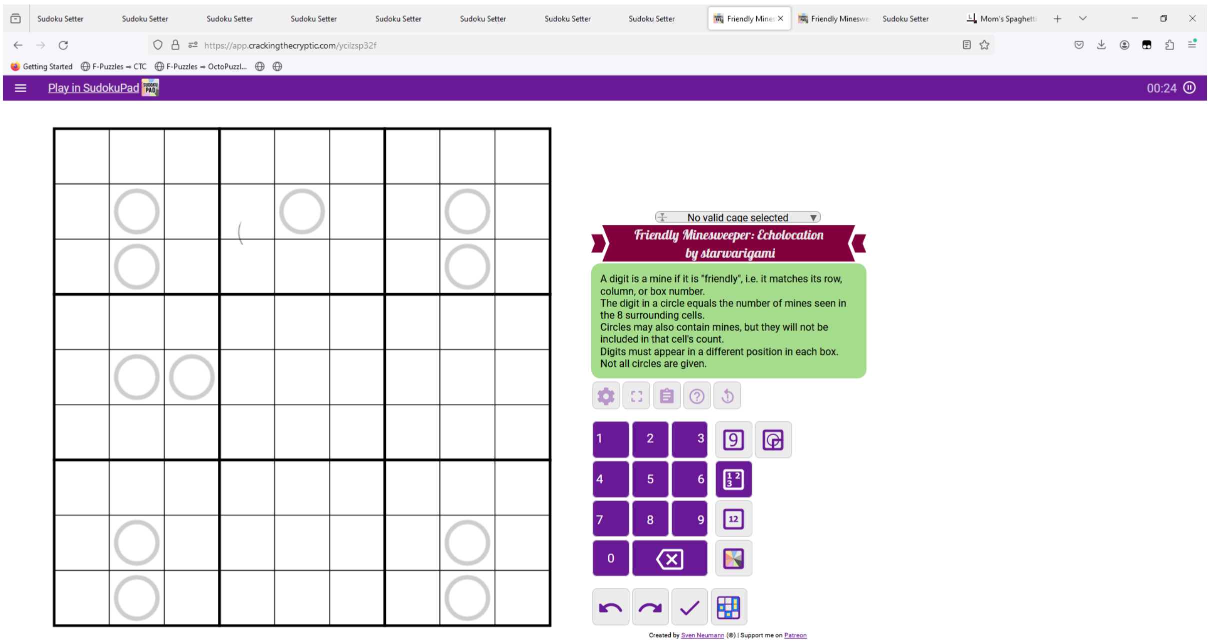
Friendly Minesweeper: Echo Location

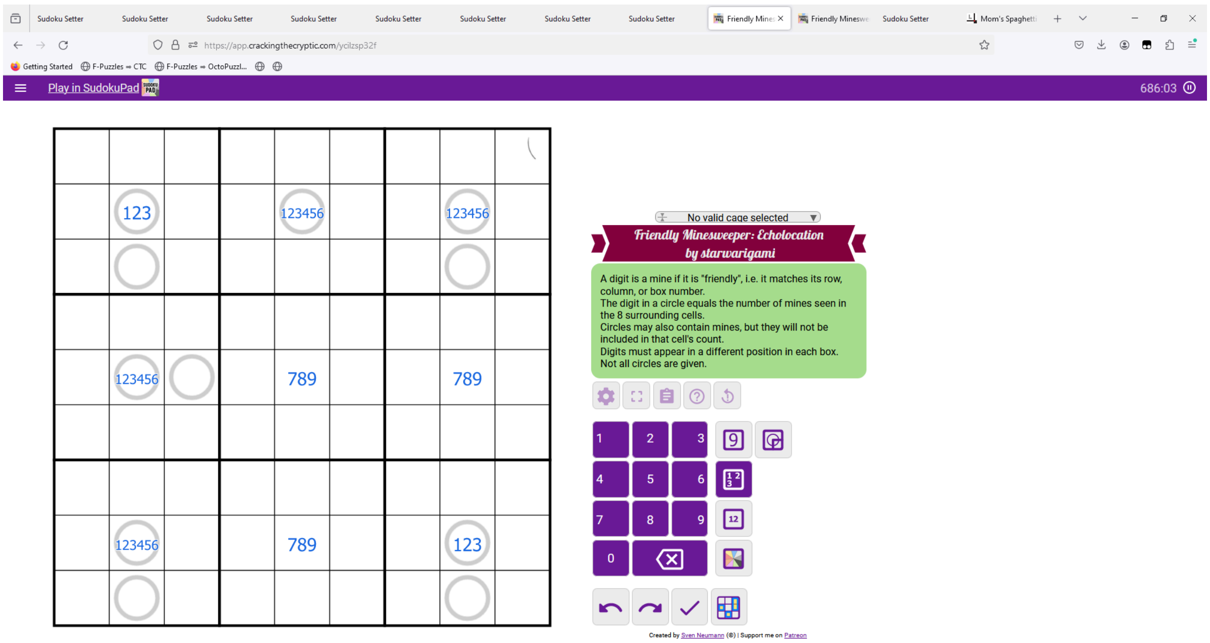


Rules

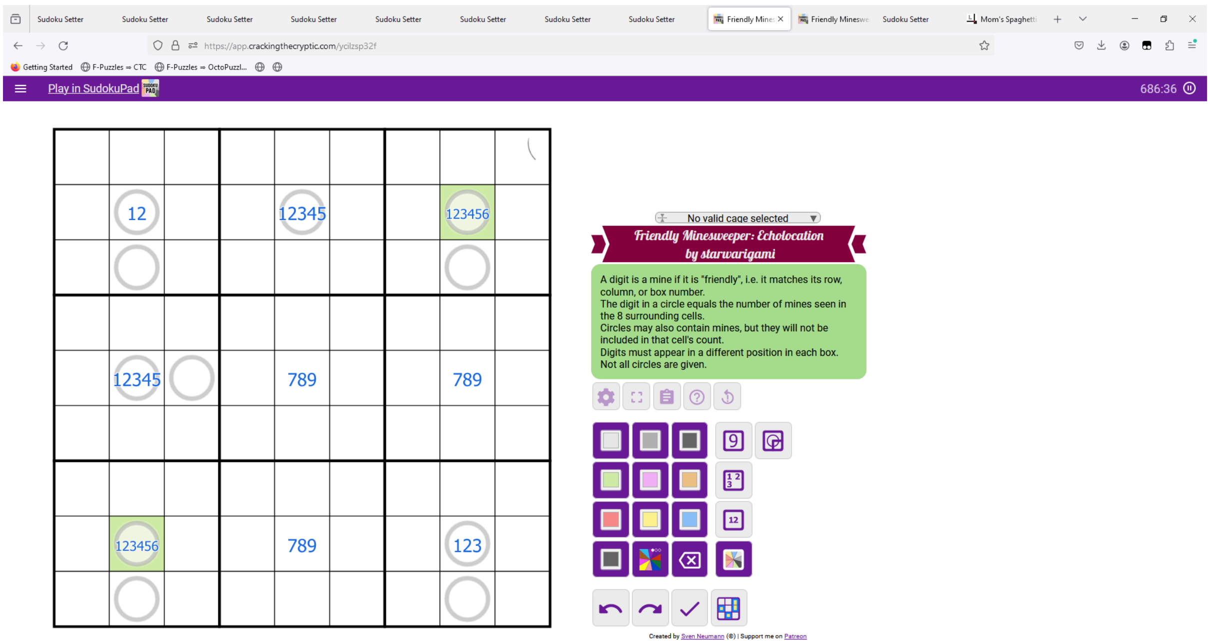
A digit is a mine if it is "friendly", i.e. it matches its row, column, or box number.
The digit in a circle equals the number of mines seen in the 8 surrounding cells.
Circles may also contain mines, but they will not be included in that cell's count.
Digits must appear in a different position in each box.
Not all circles are given.

Corollary: The maximum number of mines in a box is 6, one for each row and column passing through it. The box number always equals one of the row or column numbers. Boxes 1, 5 and 9 only have 3 possible friendly digits, as the row and column numbers are the same

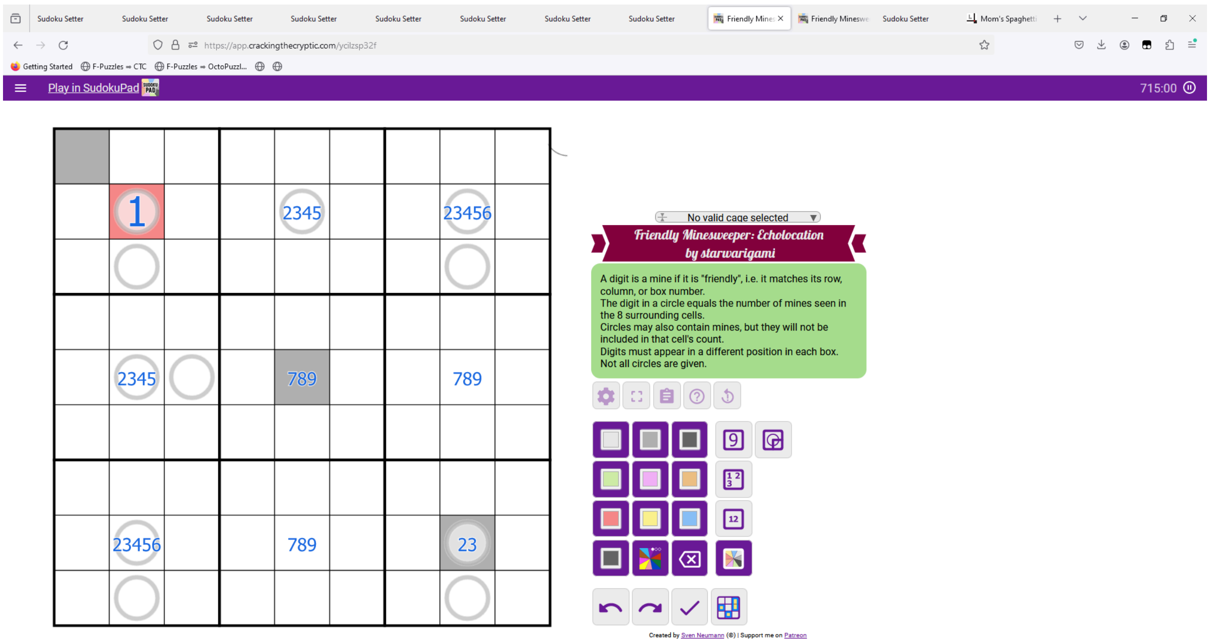
Our first observation is that we have been given circles in the middle of 6 boxes, since no box can contain more than 6 mines, by the disjoint set rule we know that these circles must contain a non-repeating set of the digits 1-6, with a 789 in the centre cells of the remaining 3 boxes. We'll pencil these digits in, remembering that boxes 1 and 9 can only contain a maximum of 3 mines.



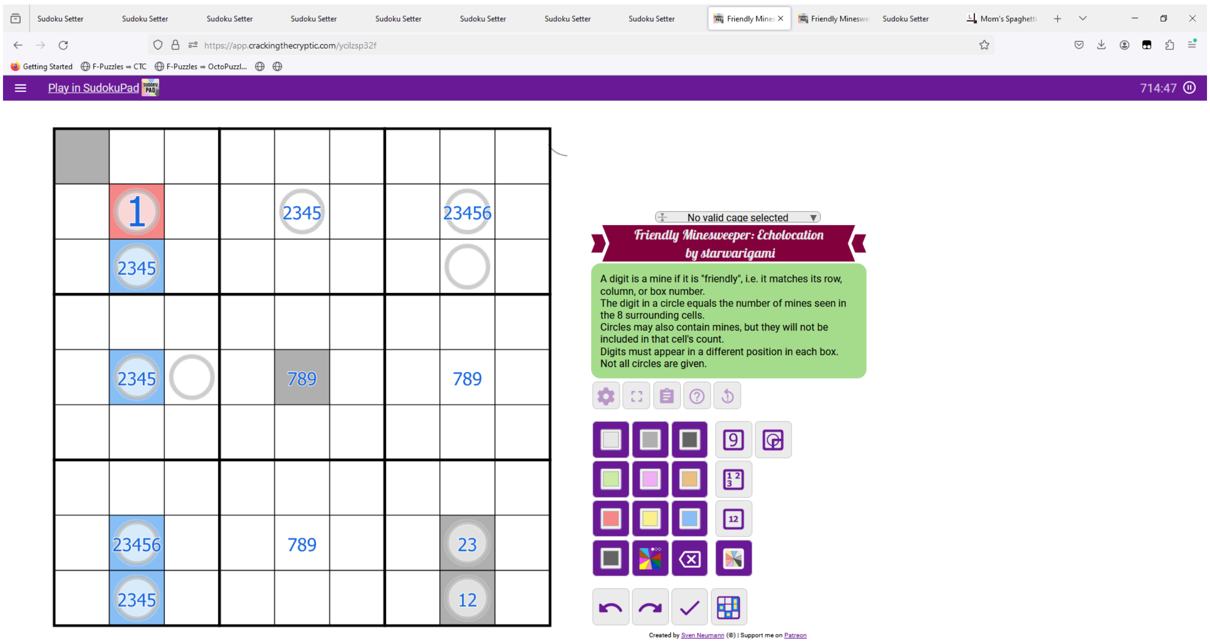
We can narrow this down further though. In order for the circle in box 1 to see 3 mines, the digit 3 would need to be in column 3 or row 3, not the centre square, so that option can be eliminated. By the same logic, we can eliminate 6 from R2C5 and R5C2, leaving only two possible locations for the 6 in that disjoint set.



6 mines in box 3 would require a 2 in row 2. 6 mines in box 7 would require a 2 in column 2. So wherever the 6 is placed, R2C2 can see a 2, so we can place our first digit, locating our first mine.

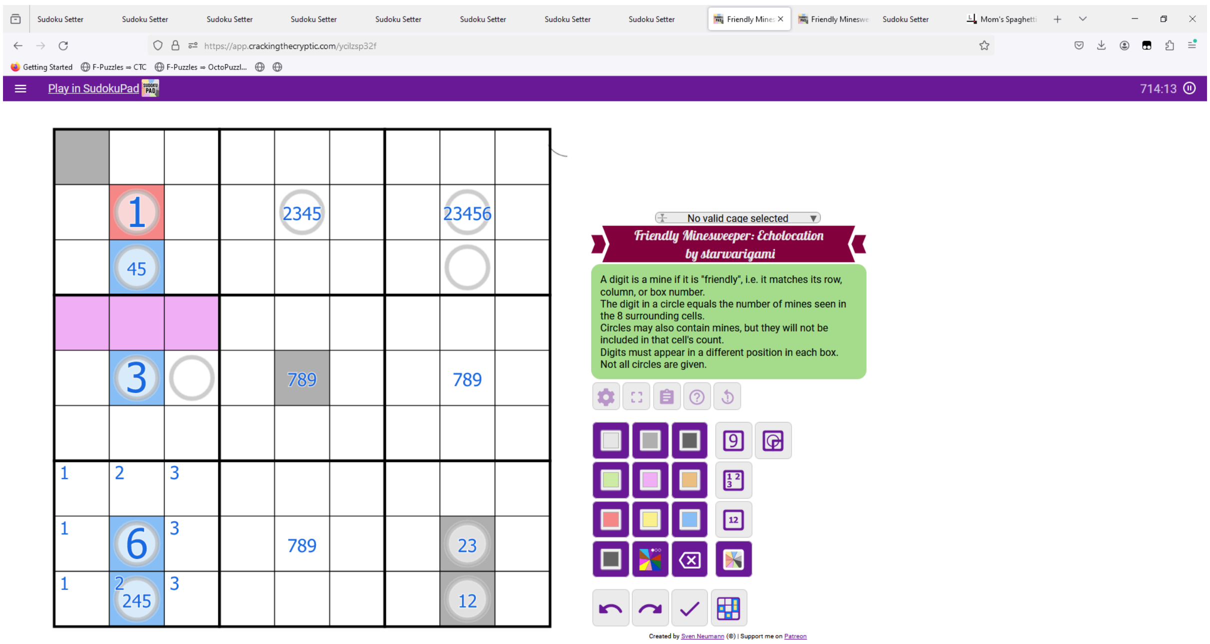


But how do we actually place the 6? Let's pencil in a few more circles. R3C8 and R5C3 aren't restricted beyond the fact that they can't contain a 9, so let's ignore them for now. R9C2 is on the edge of the grid, so only has 5 possible locations for mines surrounding it. R9C8 has the same number of squares around it, but there are only 3 potential mines in the box. It can't see all three of them though, or we would also have 3 mines adjacent to R8C8, duplicating the 3. Finally, although R3C2 has 8 squares surrounding it, we know that there can only be at most 1 more mine in the 4 adjacent unfilled squares in box 1, so this circle also has a maximum value of 5.



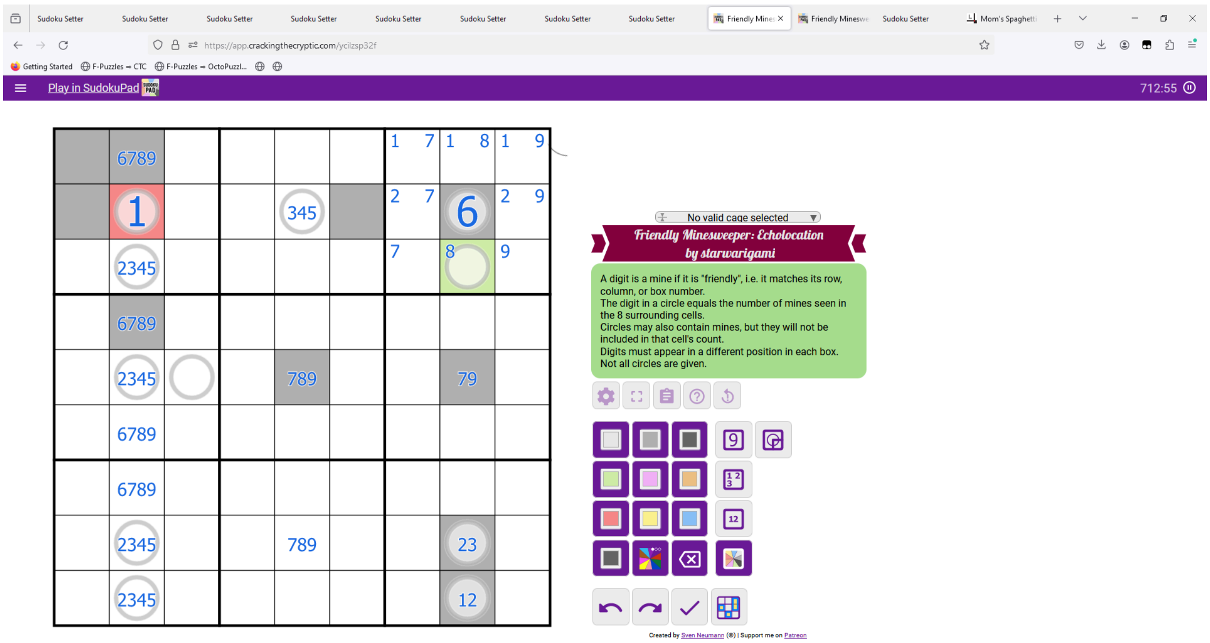
The circles in column 2 should now be catching our eye - if we could eliminate the 6 from R8C2, we would get a 2345 quadruple in the column, allowing us to get a foothold into the puzzle, so let's focus our attention here.

There are 9 possible mines in boxes 4 and 7 combined. One for each of the 3 columns and each of the 6 rows (the two box numbers overlap with one of the rows so do not contribute any more to the count). With the 6 mines in box 7, the 3 friendly digits for the columns are accounted for, knocking the 2 out of the circles in boxes 1 and 4, and placing a 3 into R5C2 as the maximum number of mines remaining in box 4.

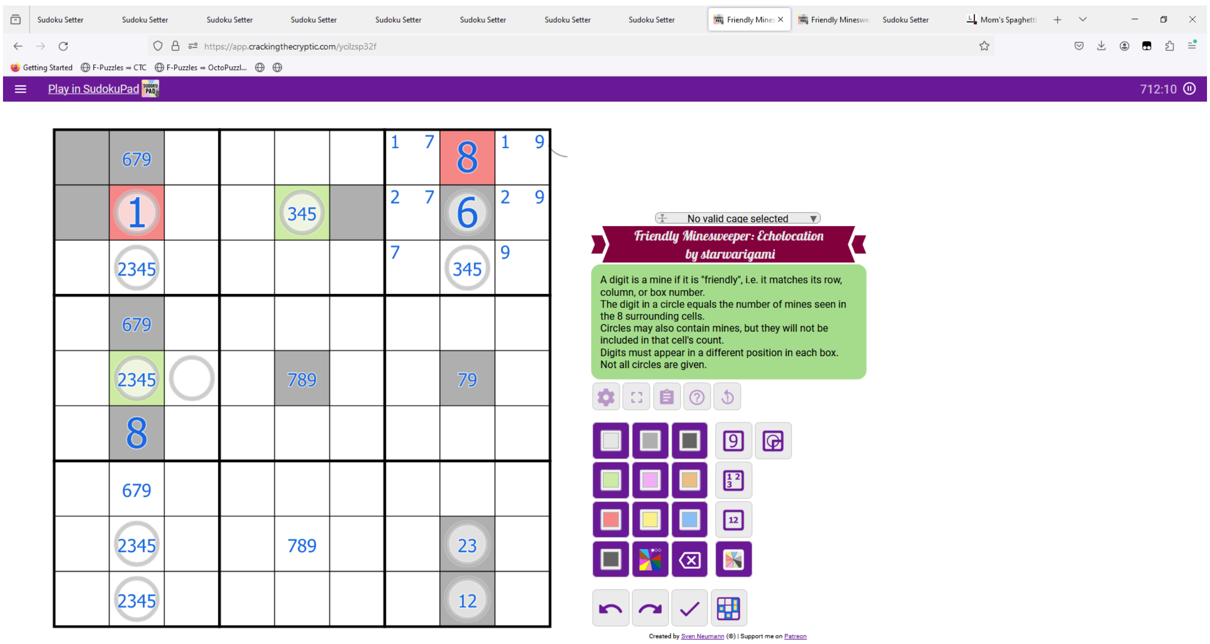


This would mean that the circle in R3C2 would need to see a minimum of 4 mines. Since there are only 2 mines in box 1, we would need at least another 2 mines in the purple cells.

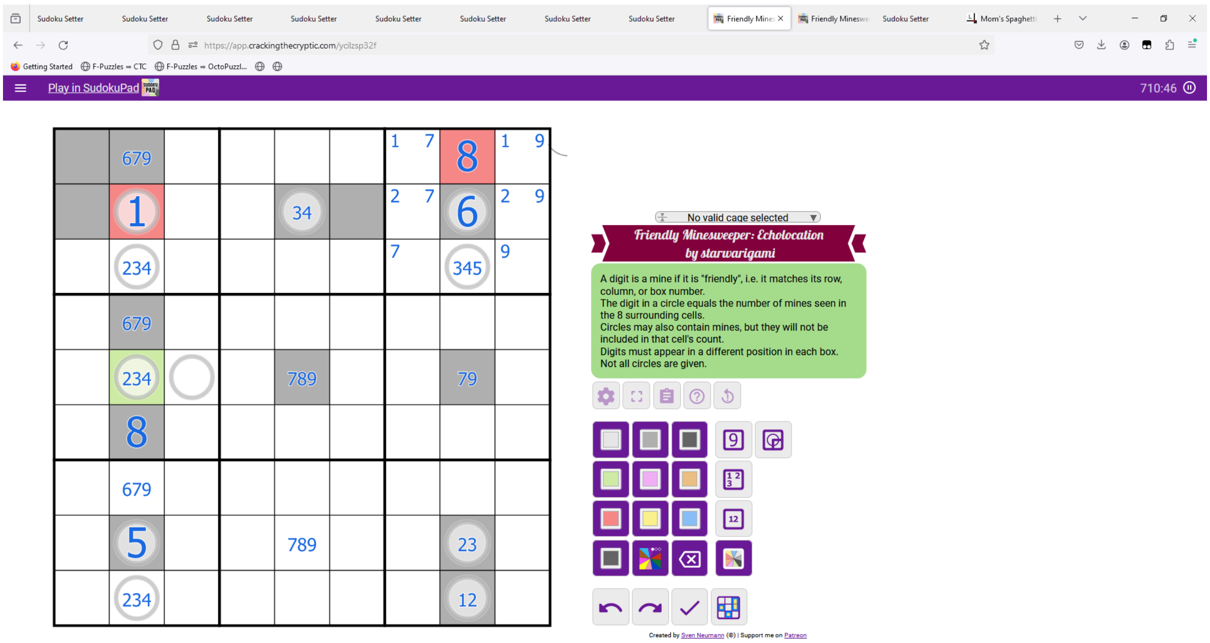
WIth the friendly column digits unavailable, this is not possible as the row number and the box number are both equal to 4, leading to a contradiction. So we can place the 6 in box 3 and pencil in our mines in that box.



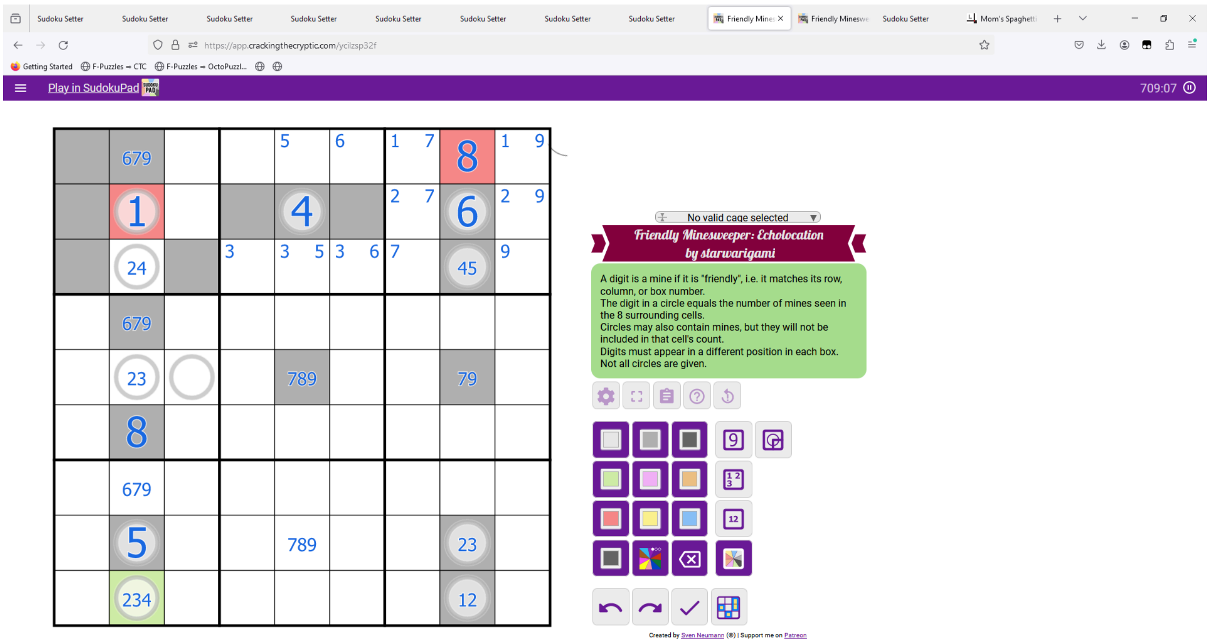
We can actually now place our second digit - Since the 6 in R2C8 is not friendly, we can't see 8 mines from the circle in R3C8, placing the 8 in R1C8. By the disjoint set rule, this also gives us the position of the 8 in column 2



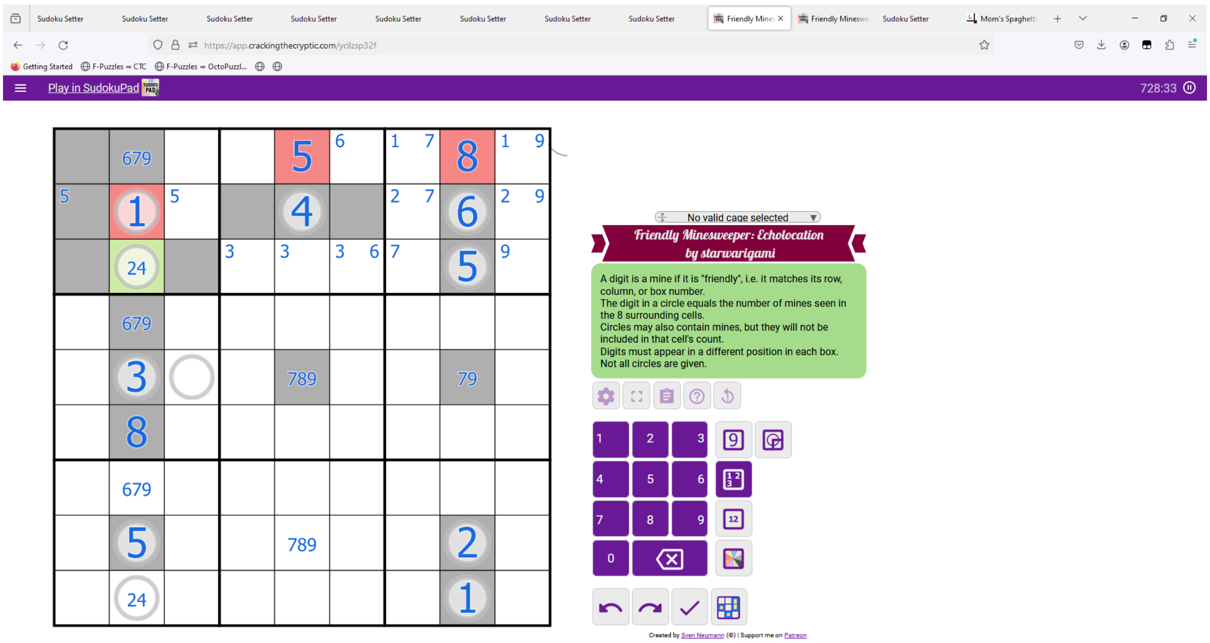
Let's do a bit of housekeeping on our circles. There are still 5 possible friendly digits in box 2, but if we placed 5 in the circle only 4 would be visible from that cell. We have the same issue in box 4. But by eliminating 5 from both of those circles, 5 must be in R8C2 by the disjoint set rule.



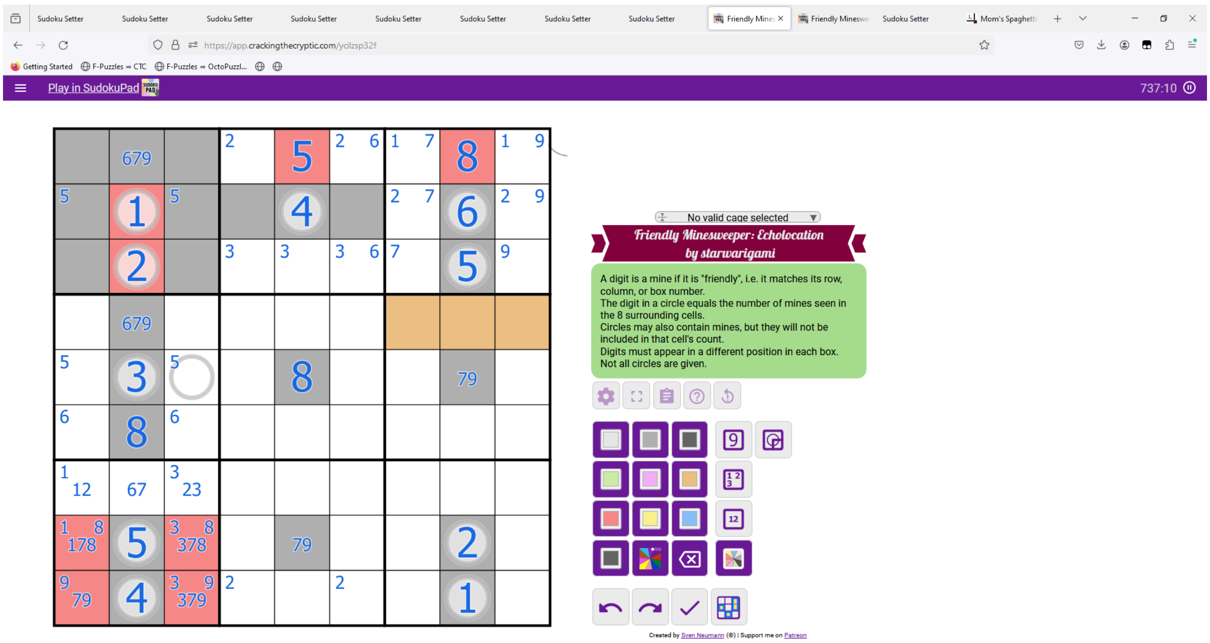
As we determined earlier, there are 9 possible mines in boxes 4 and 7. With 5 accounted for in box 7, we have 4 possible mines in box 4 - but in a similar manner as before, placing the 4 in the circle leaves only 3 mines visible from that cell. So the 4 in the set of centre cells gets place in R2C5, and we can pencil in another set of mines into box 2



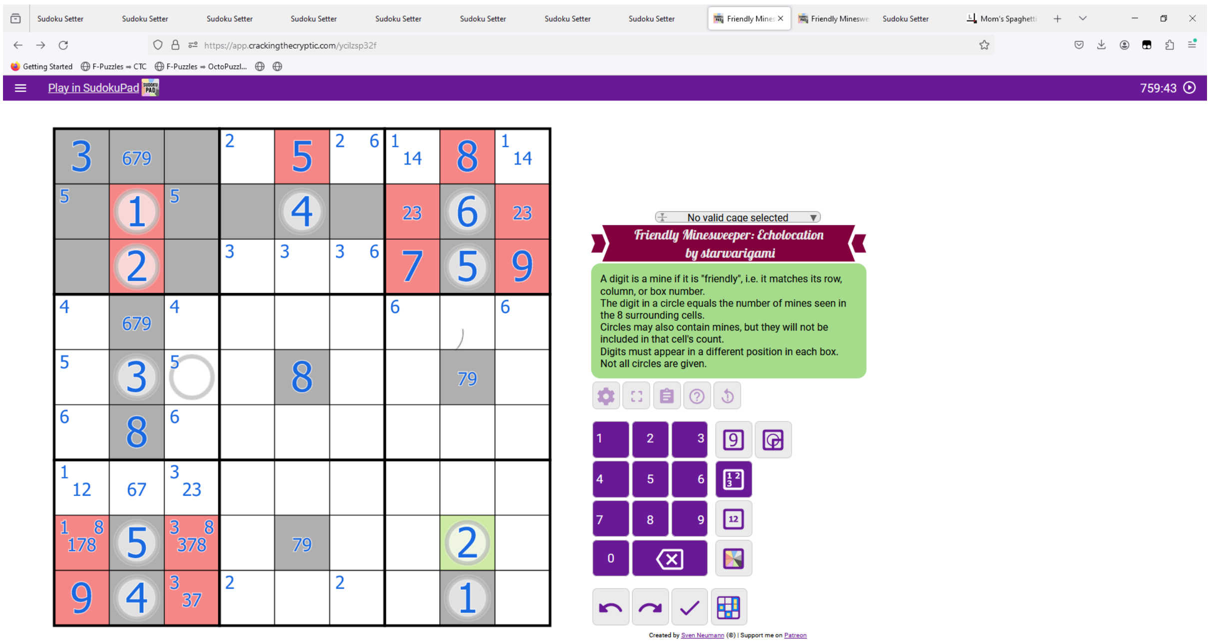
We're close to breaking this wide open now - just a couple more deductions and we'll be on our way! In order to have 5 visible mines in box 7, we need at least one of 2 or 3 to be friendly, if not both. If we were to place the 3 in the circle in R9C2, that would eliminate both possible options. By knocking out the 3, we now have a 24 pair in the column resolving the last of the two centre circles, and giving us a 5 in R3C8 by the disjoint set.



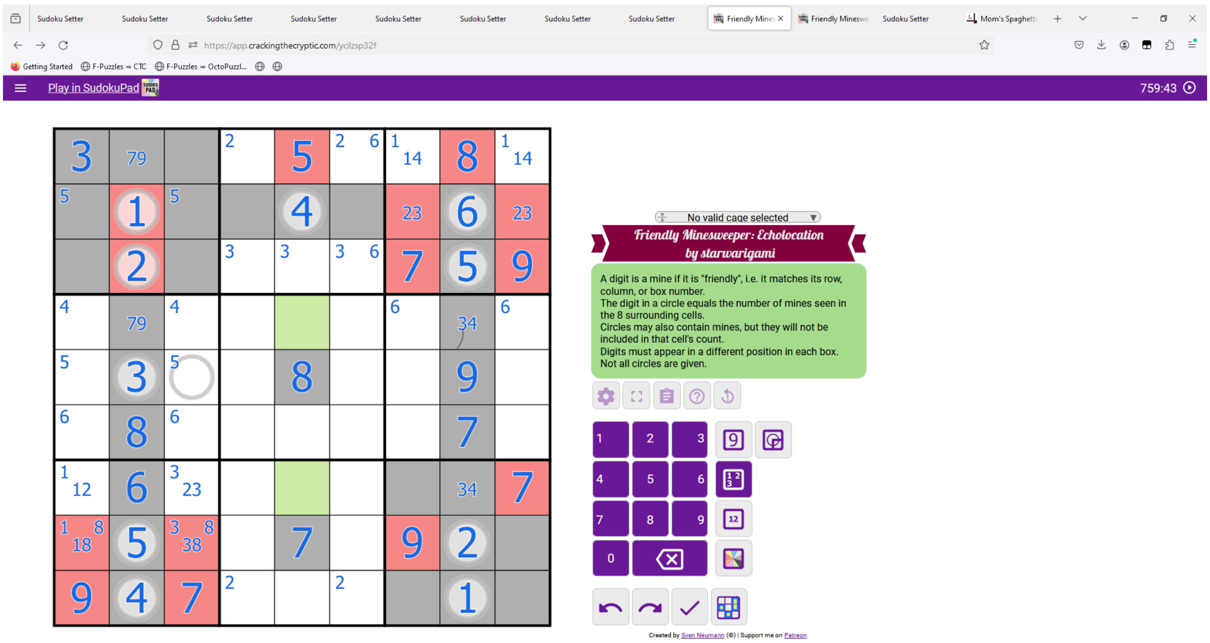
If we were to now put the unfriendly 4 into R3C2, we would need to put a friendly 3 into R2C3 to honour the circle in R2C2 using up our degree of freedom on the mines in box 7. But how then do we put two mines into the purple squares? We can't - so we resolve the 24 pair and finally blow the doors off the puzzle!



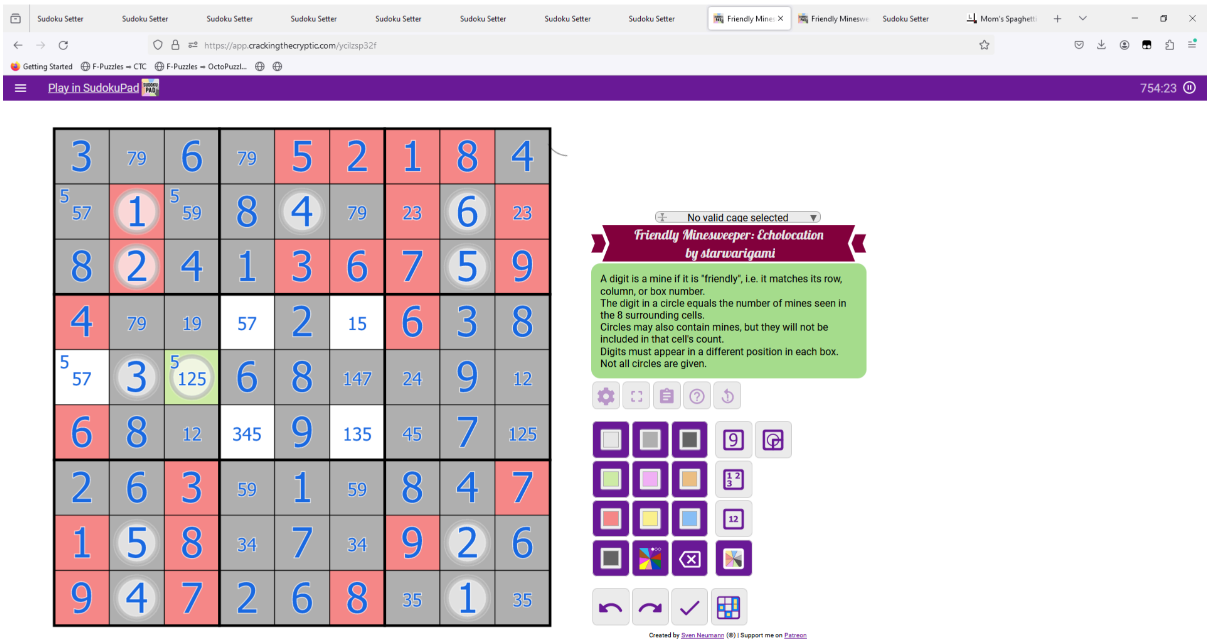
How many mines can we place in the orange cells adjacent to our 5 circle? The 4 in row 4 must be in box 4 to honour the circle in R3C2, and each of the friendly digits for the columns is accounted for in box 3. This leaves only the friendly 6 for the box number. With only one mine available in box 6, we can place the other 4 into box 3, being sure to remember to scan for cells we can update with the disjoint set rule



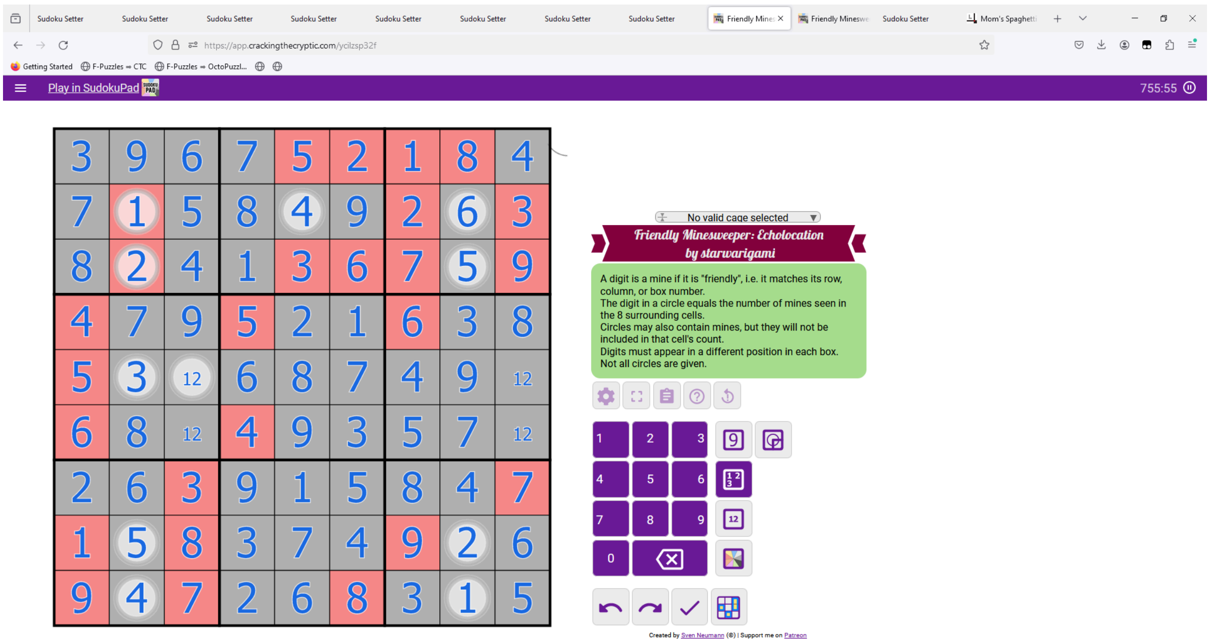
With the 8 already used in both row 8 and column 8, we need both 7 and 9 to be friendly in box 9 to satisfy the circle in R8C8. The 7 must live in row 7, so in order for the circle in R9C8 to be satisfied, we are forced with the location of the 9. The 79 pair created in column places the 7 by the disjoint set rule, and can also be used to pencil in the remaining digits in column 8



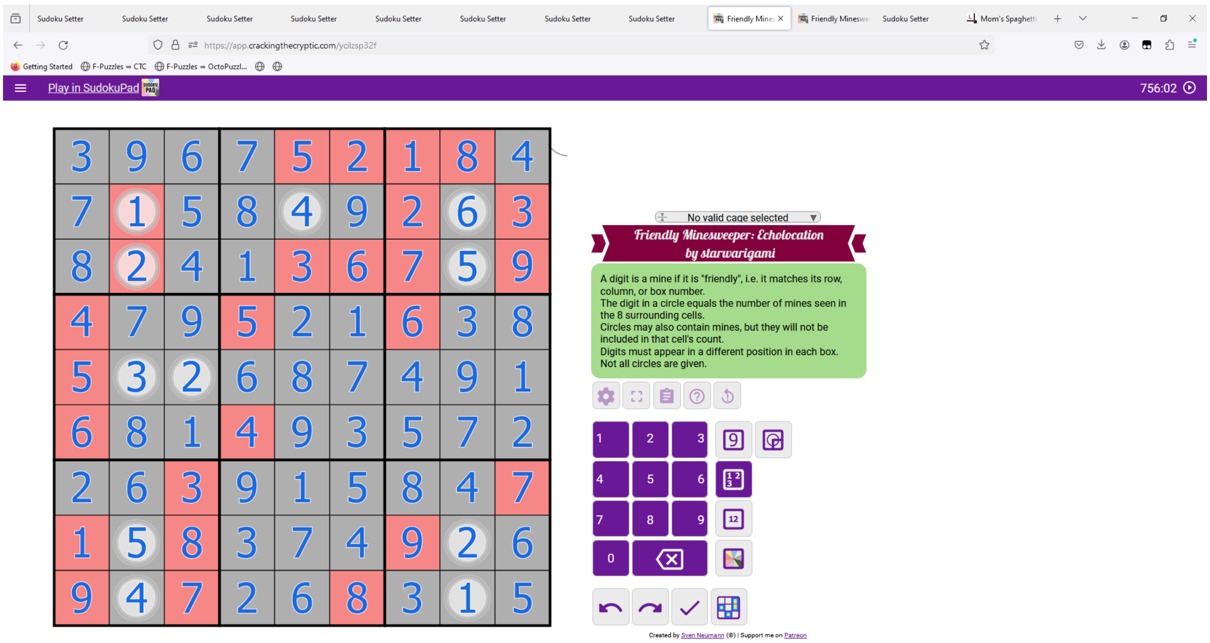
Time for sudoku - be sure to check those disjoint sets! Start by filling in the two green cells...



We clearly can't have 5 mines next to R5C3 any more...



...looks like we have 2 instead!



Final Solution