O.k., so I wish to apply some simple linear algebra principles to the following game.

You have a board of lighted buttons, and when you press a button, the lights on that corresponding row and column switch state from being lit up to being unlit and vis a vis. The point of the game is to make all the lights become lit.

To show this, let's start with a 2×2 set of buttons all turned to off. Which we represent by a 2×2 matrix of 0's (on is 1):

$$\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$$

If we hit the upper left button, then all the buttons on the corresponding row and column light up:

$$\begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$$

And if we hit the bottom right button the corresponding row and column lights switch again:

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

And if we hit the bottom left and then top right buttons:

$$\begin{matrix} 0 & 0 \\ 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

And all the lights have been lit and we win the game.

Our goal is to put forth a general mathematical analysis of this game to find a general strategy to be able to win.

Now, let's start off our analysis by identifying the sets and operation used to model this. Since the only two possible elements in any part of the matrix is 0 or 1, the underlaying matrix space should be $\mathbb{Z}_2$. Since, in general, the game could consist of any arrangement of an array, we define the elements of our vector space as an array of n elements from $\mathbb{Z}_2$ in a grid of any particular shape S, which we will call $\mathcal{M}_{nS}$. For example:

For $n = 3$

$$S = \begin{matrix} [] & [] \\ & [] \end{matrix}$$

$$\begin{matrix} 1 & 1 \\ & 1 \end{matrix}$$

would be $\in \mathcal{M}_{nS}$.

Then we define our matrix addition to add $\mod 2$ the corresponding elements in the same position of the two arrays, and we define scalar multiplication to be to multiply $\mod 2$ all the corresponding elments of the array by the scalar.

Since addition $\mod 2$ results in every element of the array being either a 1 or a 0, we are ensured closure under addition. Furthermore, we can pick up commutativity from commutivity of addition under $\mod 2$ in the corresponding elements of the array, and similarly for the existance of identity, associativity, and additive inverse.

Given an array $u \in \mathcal{M}_{nS}$ and $0 \in \mathcal{M}_{nS}$ being the identity array consisting of all elements of the array being 0, $1u = u$ and $0u = 0$ by definition of scalar multiplication; thus we've shown closure under scalar multiplication and the identity property for scalar multiplication. The laws of distributivity and associativity can be showed uncreatively by brute force.

Thus, we have satisfied all the necessary properties and we have a vector space.

This links to the game above because if we start with:

$$\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$$

And press the upper left button, this is equivalent to:

$$\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} + \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$$

Because under $\mod 2$ arithmetic, adding all of a certain row and column by 1 would switch the corresponding entries, which is how we described the game in the first place.

Therefore, our game is a subspace of our earlier described vector space. Specifically, it consists of the span of the collection of those arrays by which we can add by.

Now, as part of our goal to find a general strategy by which to beat this game, we will create a linear isomorphism to ease our analysis.

Let V_n be the vector space consisting of n-tuples over $\mathbb{Z}_2$ (such as $(0_1, 0_2, \cdots, 0_{n-1}, 1_n)$). Then we define $\phi : \mathcal{M}_{nS} \to V$ to be a mapping that takes each element of an array in $\mathcal{M}_{nS}$ and according to its position in the array, put it in a unique position in the n-tuple. The fact that V_n and $\mathcal{M}_{nS}$ both consists of vectors with n elements ensures onto-ness, and that it maps the positions uniquely guarentees one-to-one-ness.

It is a linear transformation since:

ϕ(array of elements labeled a + array of elements labeled a)

= ϕ(array of elements labeled a + elements labeled b)

= n-tuple of elements labeled a + elements labeled b

= n-tuple of elements labeled a + n-tuple of elements labeled a

And

$\phi(1u) = \phi(u) = 1\phi(u)$

$\phi(0u) = \phi(0) = 0\phi(u)$

Thus scalar multiplication and vector addition operations are preserved.

Here's an example of what the isomorphism does:

$\phi(\begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}) = (1, 1, 1, 0)$

(Although $(1, 0, 1, 1)$ would be just as good, so long as the position to position map is consistent then it's fine)

Thus we can view this as the more easily envisioned n-tuples.

Now let's go back to our original subspace of $\mathcal{M}_{nS}$ in which our game is being played. We want a linear combination of our allowable vectors in order to get the all-1 array.

Or, equivalently, we are given a set of n-tuples, $\{v_1, v_2, \cdots, v_n\}$ (note: yes, this n is the same as the number of elements in the array, and it is justified since we have a distinct vector which we add by that corresponds to a distinct push of a button on the board) and wish to find the linear combination to get the n-tuple consisting of all 1's, $(1_1, 1_2, \cdots, 1_n)$. This process is well-known and to remind the reader, we apply Gaussian elimination to the following (the fact that we are in $\mathbb{Z}_2$-world does not change this operation):

$$(v_1^T v_2^T \cdots v_n^T | (1_1, 1_2, \cdots, 1_n)^T)$$

And can find the appropriate linear combination or can tell if it is even possible by looking at the resulting augmented vector.

In summary, apply the ϕ-isomorphism to the corresponding n-vectors, and perform a Gaussian elimination with these vectors in order to win the game.