**Project Plan**

**Students:**

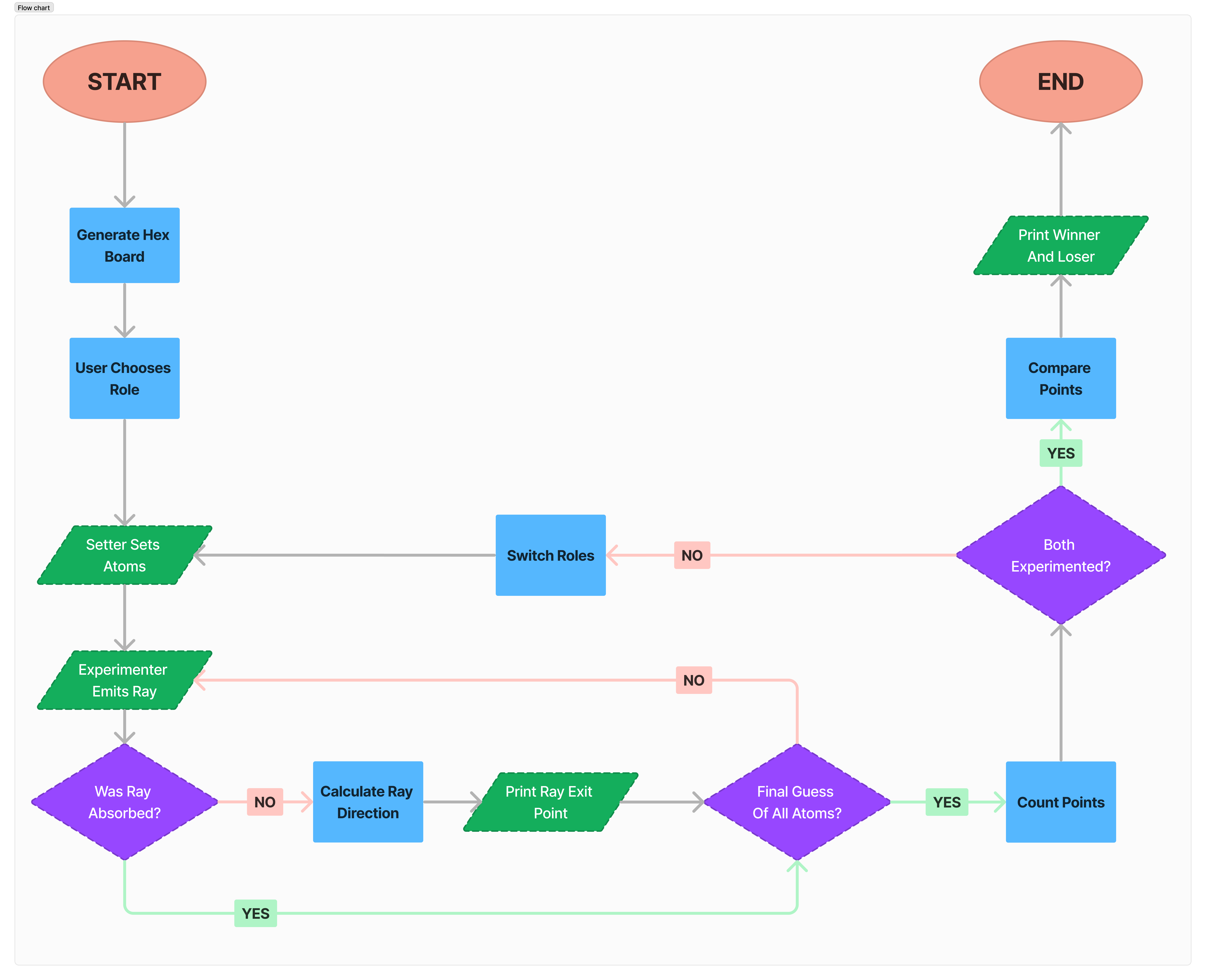
Liam Kerrin - 22371791

Narcis LASTNAME - NUMBER

Abdullah Shinwari - NUMBER

**Software Architecture and Logic**

In this brief section we will cover what we did in planning the logic or the game. Here was obviously a need for a clear gameplay loop, and a series of steps involved in making this occur, so we constructed a flowchart in a program called FigJam (<https://www.figma.com/figjam/>). Here it is below, then an explanation will follow:



A high-definition image should also be available on GitHub repository, in:

Documentation/img/flowchart\_1.png

The circular shapes are start/end, the rectangles are logical calculations for the program to do, the parallelograms are input/output, and the diamonds are branches.

Essentially, the game would begin and generate a visual for the hexagonal board, then the user would choose the role. Once the role has been chosen, there is an opportunity for the setter to place the atoms (this could be either the user or the computer depending on choice of roles previously mentioned). Once the atoms have been placed, it is the experimenter’s turn to begin playing. This then begins the main gameplay loop, where the player can continue placing rays until they are ready to make an educated guess for the placement of all the atoms. Once the guess has been made, points are tallied for both user and computer (based on their current respective roles), then the roles are switched, and the gameplay loop begins again.

It is important to note that the points are tallied before the switch, because the state of the game will be reset when the roles are switched, so we use the current data to add points to the user and computer before they switch roles. Finally, once both the user and the computer have played both roles, the program compares points, and prints the winner (the one with the least points overall) and the loser, then the program ends.