How the movements are handled

: Before anything happens, the player will always be passing its own location to the Maze and get all the possible directions in which the player can move. Then it will check to see if the user-wanted move is among the legal moves that were received. If it’s not legal, the user will be told that they can’t go through walls. If the move is allowed then, the coordinates of the player will be updated and a circle worth of cells around the player will be revealed. Then the game will check to see if the player has run into any cats or the cheese. In terms of cats moving, their randomly generated movement will be verified through the same process as the player but as the last step, their previous coordinate will also be known as an only-cross-if-necessary, so that the cat can make its random movement.

How the area around the player being revealed is handled.

The player has a location of x and y coordinates. The maze itself also has a 2d array of “cells” that also has a location of x and y coordinates. The “cells” also hold states of whether a certain location is either a wall or an open space, and whether it is revealed or not. When the player moves, its new location is then sent to the maze class. There the maze class goes to a cell on the 2d array using the coordinates as indices on the 2d array. There the maze then sets all cells around the certain cell, changing all the states around the certain location to being true. So it changes cells with locations with coordinates X-1 to X+1 and Y-1 to Y+1, and changing their state to revealed. This makes it so that the player only passes onto maze what the maze needs to know. This would all be communicated between the game class.