I organized my program around two enumerations. These enums were a direction, which defines where pacman will move and a GameObject which defines the different entities that exist in the game. I used a two dimensional array to store the game state in which each array value is a game object. This can directly translate to the output since the game board is also a grid. The initial board is created by randomly assigning the array values a cookie or empty value. The game spec wanted an 8% cookie allocation, so I used a random number generator to provide this ability. The result is that sometimes there will be a few more or a few less cookies than you would expect. It is especially obvious when using small board sizes because the lack of spaces results in many boards having no cookies at all. The main flow of the program is to draw the current state of the game board followed by prompting the user for a command and finally executing that command.

I considered representing game objects by strings or integers, but the problem always arose that following what objects represent what gets convoluted when they are simply numbers or strings. One thing I would do differently is provide better handling of initializing cookies. The random number generator works well enough over time, but it struggles when the board is small. Perhaps adding logic that for boards smaller than a certain size we guarantee a certain number of cookies to exist, rather than relying on chance.