Eli Landa,
Matt Rudenko

Cat and Mouse Game OOD:

1: Position Manager - Update Maze
1. Read in the destination the player is trying to go
2. use Input validation class to check if its a valid move
3. if it is a valid move, set the BoardObject tagged as player to be tagged as a path and set the destination index to be tagged as the player
4. if it is not a valid move return the gameboard with no changes made
5. search through the maze array for the BoardObject tagged as cat.
6. randomly pick a direction for the cat to move to that is different from the one it was just in previously.
use the inputValidation class to check if the cat can move there. if not re pick. if it can, set the BoardObject at the position where cat currently is to path and set the destination index's BoardObject to be tagged as cat
7. if no possible paths are found that wasn't in the direction that the cat had already come from, back track and move the cat back in the direction it came from.
8. check if there are any BoardObjects tagged as cheese in the maze. if there are not, pick a random BoardObject tagged as a path and set it to be tagged as cheese. increment amountOfCheeseCollected by 1
9.

2: Maze Builder
1. Creact a 2 dimentional array of BoardObjects
2. set all objects in the array to be walls and invisible
3. Set the BoardObjects in the [0][Y], [maxX][Y], [X][0], and [X][maxY] (xMax and yMax, are the last indexes) indexes to be visible. this will make the outer boarder of the maze visible
4. Select a random starting point on the array. Use odd numbers for row and column. Set the BoardObject at this index to be path instead of wall
5. use x and y variables to keep track of your position
6. randomly pick a direction to go (either up, down, left, or right) - Before finalizing direction,
check if two cells into that direction is outside of the bounds of the maze (needs a valid array index) and make sure that the path in that direction is a wall.
if those conditions are not met, keep re picking directions. if it finds that it is at a dead end, we need to backtrack (more later).
7. once the direction is picked, always move 2 cells in that direction and set those cells that you traversed to be paths and not walls
8. If a dead end is found, backtrack along the path until a new direction that meets the requirements layed out in step 66 is found. if none are found the maze is complete
9. traverse the 2dimensional array that this maze is stored in looking for BoardObjects tagged as walls. pick a random number between 1

and 10. if the number is less than a 3, re tagg this BoardObject to be a path.

10. Set [1][1] to be tagged as player

11. traverse the dimensional array that this maze is stored in and create an array of all the indexes containing BoardObjects tagged as path. randomly select from these indexes 1 index.

12. set the BoardObject at that index to be tagged as cheese