Developer Documentation

Code Structure and Organization

Position Class:

- The Position class represents a 2D point and provides functionality for movement and manipulation. Its purpose is to enhance ease of use and functionality within a grid system.
- The constructors initialize the object with default values, with user-defined coordinates, or with user-defined coordinates within a specified grid size.
- Accessor methods such as getX, getY, getXMax, and getYMax retrieve the values of private member variables.
- Mutator methods (move, reset, and rehome) enable users to modify the position, reset it to its home position, or update the home position, respectively.
- The equality operator (operator==) compares two Position objects to check if they are equal.
- The method for displaying prints a visual representation of the grid with the current position marked, while also showing the home position.

Position.cpp:

- The Position class implementation file has definitions for the methods declared in the Position.h file.
- The move function changes the position according to a specified direction (1-9).
- The reset function returns the current position to the home position.
- The rehome function updates the home position to the current position.
- The updateHome function checks if the current location is outside of the grid and updates the home position accordingly.

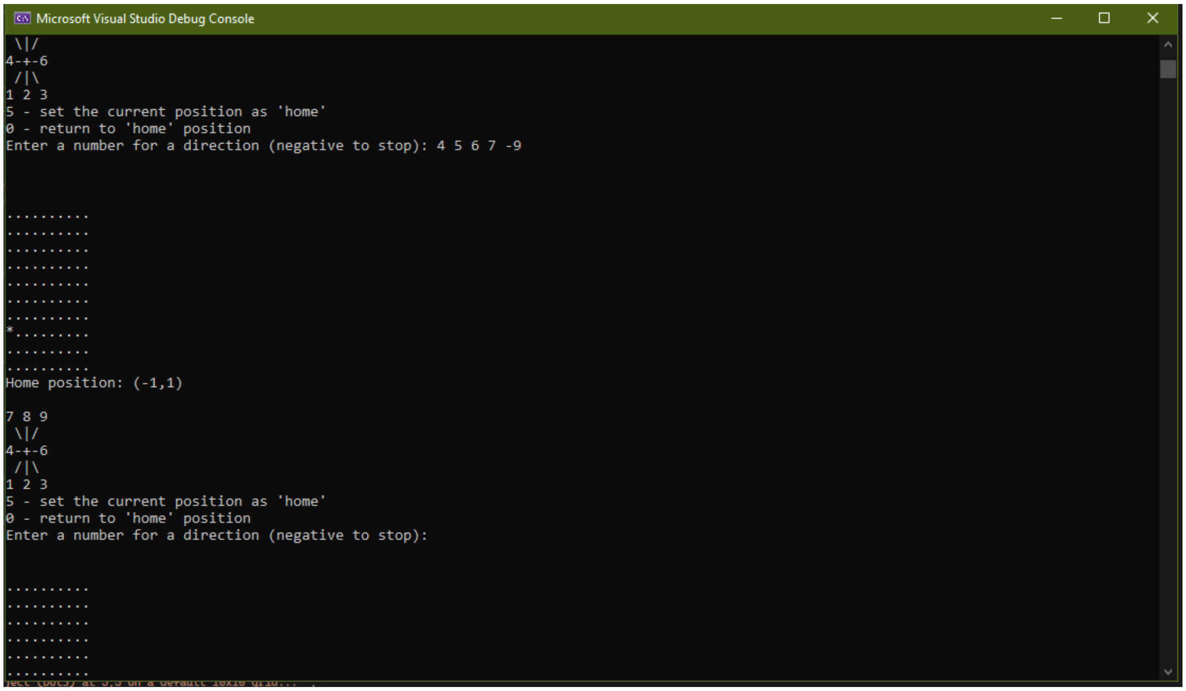
Main.cpp:

- The main program file exemplifying the use of the Position class.
- The functions comprise getDirection to gather the user's input for movement direction and botNavigation to execute bot navigation according to the user's input.
- Testing is performed for various scenarios, including constructor usage, position comparison, and bot movement.

Overall Usability:

- The program is user-friendly, thanks to its modular design, well-commented and readable code, effective error handling, extensive testing, clear user interaction, visually informative grid displays, consistent coding style, and minor recommendations for improvement, such as managing edge cases and adding more comments for clarity in the getDirection function.

Examples Program Running:



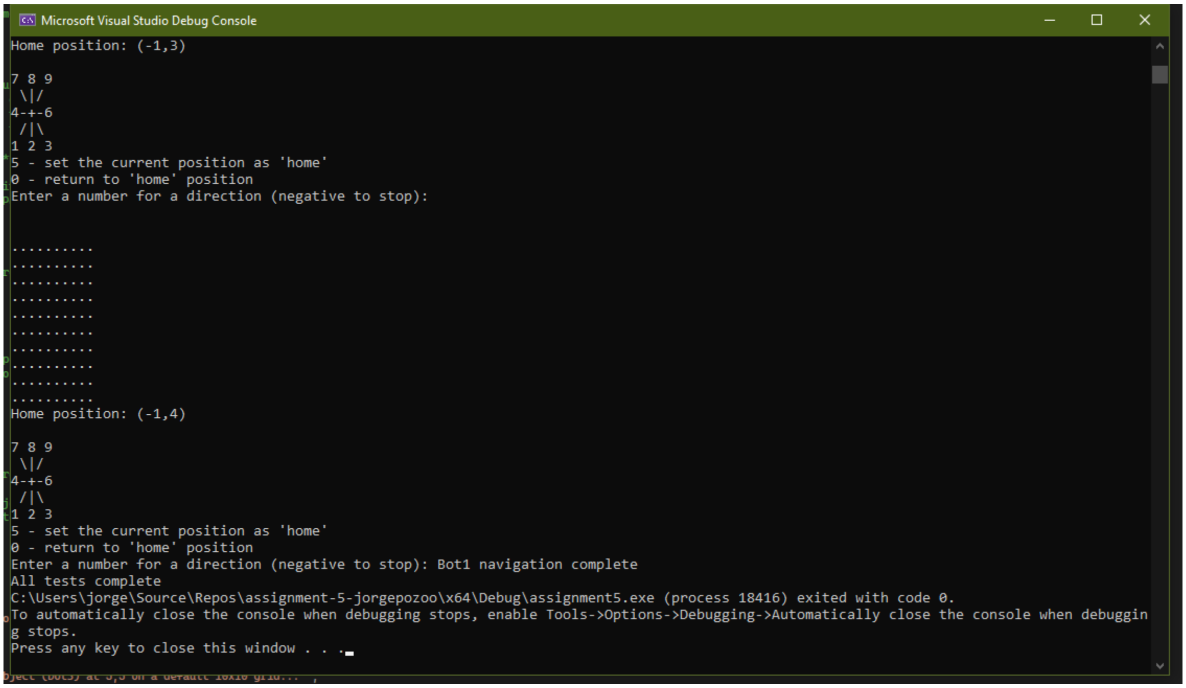
```
 \|/
4-+-6
 /|\
1 2 3
5 - set the current position as 'home'
0 - return to 'home' position
Enter a number for a direction (negative to stop): 4 5 6 7 -9



..........
..........
..........
..........
..........
..........
..........
*.........
..........
..........
Home position: (-1,1)

7 8 9
 \|/
4-+-6
 /|\
1 2 3
5 - set the current position as 'home'
0 - return to 'home' position
Enter a number for a direction (negative to stop):


..........
..........
..........
..........
..........
..........
```



```
Home position: (-1,3)

7 8 9
 \|/
4-+-6
 /|\
1 2 3
5 - set the current position as 'home'
0 - return to 'home' position
Enter a number for a direction (negative to stop):


..........
..........
..........
..........
..........
..........
..........
..........
..........
..........
Home position: (-1,4)

7 8 9
 \|/
4-+-6
 /|\
1 2 3
5 - set the current position as 'home'
0 - return to 'home' position
Enter a number for a direction (negative to stop): Bot1 navigation complete
All tests complete
C:\Users\jorge\Source\Repos\assignment-5-jorgepozoo\x64\Debug\assignment5.exe (process 18416) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debuggin
g stops.
Press any key to close this window . . .
```