# 1. Description

Your job is to create a robot simulator. The robot moves around a 30 x 30 grid floor. The robot has a pen that has two possible positions up or down. When the pen is in the up position the robot moves around that floor without leaving a mark. When the pen is in the down position the robot leaves a mark as it moves on the floor. The robot can move in four directions: **north, south, east, and west**. Ensure that the interface with the user is easy to understand and use. When the robot reaches an edge it will wrap around to the opposite edge.

# 2. Specification

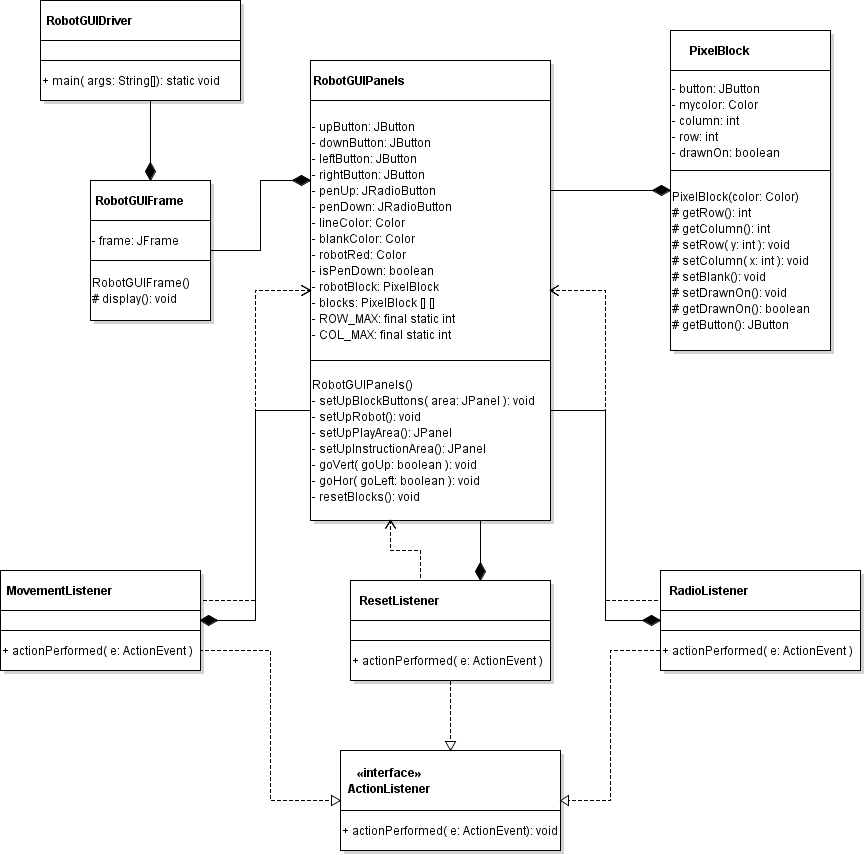
* GUI
* One player
* 30 x 30 grid
* Cardinal direction movement
* If movement spaces go over available spaces, wrap to other side
* Robot has the ability to move while drawing or move without drawing

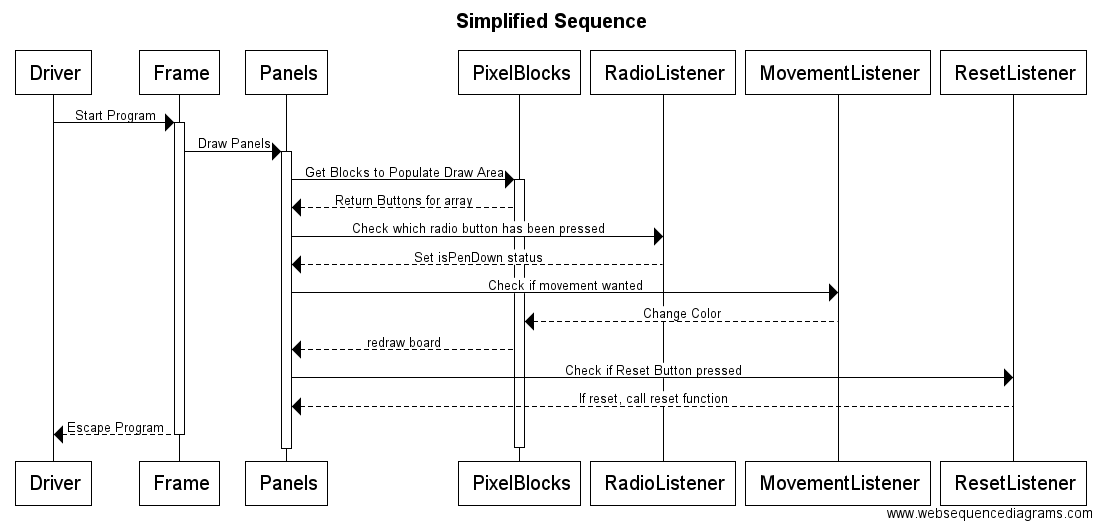
# 3. Dependencies

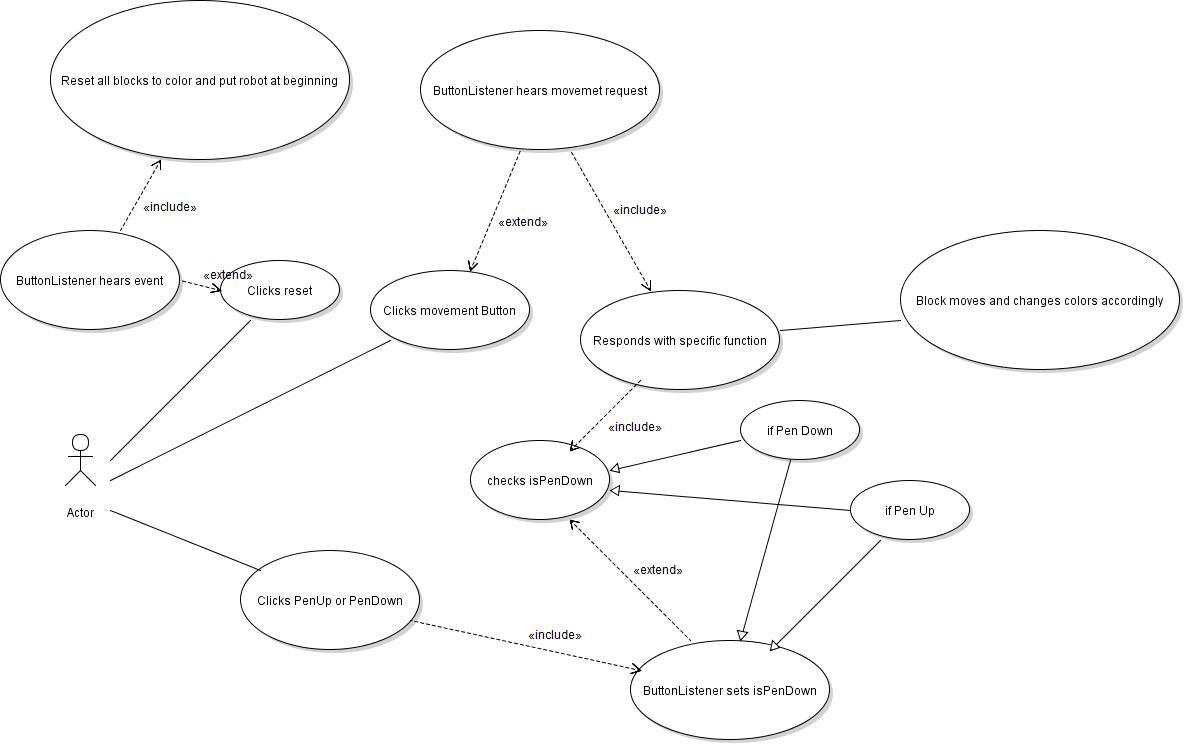
Program calls for java compiler with visual java display.

Class Dependencies: Program has linear dependency. Driver calls on the frame, which calls on the panel which is composed of draw blocks. Panel uses interior classes for Action Listeners (separated into three classes, despite all being action listeners, so that they have single responsibility). The Pixel Block class COULD be used on its own, though being comprised of GUI components, it needs a Panel of some sort.

# 4. Design







# 5. Test plan

|  |  |  |
| --- | --- | --- |
| Player Class | | |
| Test# | Requirement | Test |
| 1. | Does the program Start | Start GUI program |
| 2. | Does the frame draw | Does it display anything |
| 3. | Do the panels draw | Does it fill the display |
| 4. | Should not be resizeable | Cannot be dragged to resize |
| 5. | Text is presented clearly | Text can be read, buttons legible |
| 6. | Buttons press | Interactable buttons have click event |
| 7. | Clicking on the play area does nothing | No click event in play area |
| 8. | Robot block starts at 14 x14 | Yes |
| 9. | There is a grid of 30 x30 | Yes |
| 10. | Radio buttons are inclusive (only one can be clicked at a time) | Yes |
| 11. | Robot can move around in 4 direction | Red block moves when arrows are clicked, one movement per click |
| 12. | If robot goes over edge, shows up on the other edge | Yes |
| 13. | Robot can draw | Radio button “Pen Down” and movement means grey turns to black per unit moved. |
| 14 | Switching to pen up means no drawing | Continued movement with pen up produces no color change |
| 15. | If a drawn line is gone over with pen up, it keeps the line. | Yes. |
| 16. | Reset resets everything back to original except for radio button selection | Yes |
| 17. | PixelBlock requires a color for creation | Yes |
| 18. | You can set the row and column of a PixelBlock (not setting it comes up with Null) | Yes (setRow(int y), setCol(int x)) |
| 19. | You can return the row and column (comes up with null if not set) | Yes (getRow(), getCol()) |
| 20. | SetBlank() sets the drawing status and color back to grey | Reset function performs correctly |
| 21. | setDrawnOn() sets the pixelblock isDrawnOn to true | When drawing over lines with pen up, it doesn’t reset it to base grey |
| 22. | getDrawnOn() returns whether block has been drawn on | See above |
| 23. | getButton() returns the button | playArea would be blank if it didn’t work. |

# 6. Source Code

[Softcopy only]

# 7. JavaDoc

[zipped together with this document]

# Revisions

|  |  |
| --- | --- |
| Date | Brief description |
| 11/02/2016 | Tore apart console version…basically threw it all away and plotted out GUI version |
| 11/07/2016 | Made base UML and sequence diagram |
| 111/09/2016 | Setup GUI, created PixelBlock class, reworked UML to show changes |
| 11/12/2016 | Found a problem with my movement (drawing over lines), fixed it |
| 11/13/2016 | Changed GUI color and further specialized classes and made things more legible |