*Systems Programming Assignment 2 Development Diary*

| Date of Session | Time Spent | Development Segment | Notes |
| --- | --- | --- | --- |
| 10/11/2021 | 1 Hour 20 Minutes | Setup | Set up a private git repository for the assignment, using the cloned data from wrippin/xv6-assessment. Then set up an authentication token to allow easy interaction with the repository via the Ubuntu VM command line. Tested the commit/push/pull commands to ensure they all worked as expected on the device – to prevent any future issues arising with these functionalities. |
| 10/11/2021 | 3 Hours 30 Minutes | Stage 1 | Implemented the basic clearing and pixel setting functions. Issues still arise in which the screen is not initially cleaned when the graphics mode is entered, and there are efficiency issues that may become relevant later, however the basic functions required for drawing are present. |
| 12/11/2021 | 2 Hours 50 Minutes | Stage 1 | Implemented the move position function to allow for changing of the graphics cursor position. Additionally implemented the “line to” functionality using the Bresenham line algorithm. These two functions as well as the example commands provided in the specification have been implemented in the “Painter” command, which as of yet takes the following arguments:  Painter [“-e”,”-p”,”-l”]  -e: uses the example square drawing functionality  -p [x] [y]: Plots the points specified in arguments x and y  -l [x0] [y0] [x1] [y1]: Sets the movement position to the arguments provided in x0 and y0, then draws a line to x1 and y1.  As of yet these functions have no error testing built in, nor any hdc usages. |
| 12/11/2021 | 1 Hour 20 Minutes | Stage 1 | Modified the existing arguments for the painter command as such:  Added the -m argument to allow the movement of the cursor towards an arbitrary coordinate  Changed the -l argument to no longer allow movement position to be set beforehand, this function is now expected to be called after the -m function  In addition, changes were made to ensure that the movement values had correct default values, and capacities were applied to user inputs – all inputs have been given a minimum value of 0 and a variable maximum based on the value it represents (319 for x coordinates, 199 for y coordinates). Finally, some text was appended to explain the existing usages in the case of an invalid input, and all functions were given a minimum number of arguments. |