**Analysis Outline:**

Intro:

* Schools enforce the documentation of coding projects and homework assignments, though various places online say that documentation is either barely there or non-existent
* Our study wishes to understand why comments may or may not be written
  + As well as general feelings of comments by subjects, good or bad
* Similar to a study conducted by Sam Wilmot at Balisage titled  *Literate Programming: A Case Study and Observations*
  + Their focus however was on a much wider scale
  + Their study draws a conclusion that commenting is too costly – more work to be done, more time used

Justification:

* This study can be defined as a case study as it is attempting to find **a link through observation** rather than a huge amount of data gathering like a survey

Design:

* Type of Case Study: Causal
  + Links various factors as to “Why” programmers don’t comment
    - Asks for factors such as current language and IDE
      * Main idea was that certain IDEs could cause the user experience to be unpleasant and reduce wanting to comment
    - Why they are doing the project
    - Where the code was first given
      * Supplied code could include premade comments from supplier, reducing drive/need for adding ADDITIONAL comments
    - Determining external reasons for not commenting
      * Similar point as to if the code was premade/give, determine the possibility if reasons exist for not commenting
        + Time – deadlines possibly playing a key factor, causing comments to be seen as extraneous over actually getting the program to work
        + Space – Some scripts can amount to 100’s of lines of code, adding comments may be considered a clutter
        + Definition – what does the current code mean

Not easy to put into words

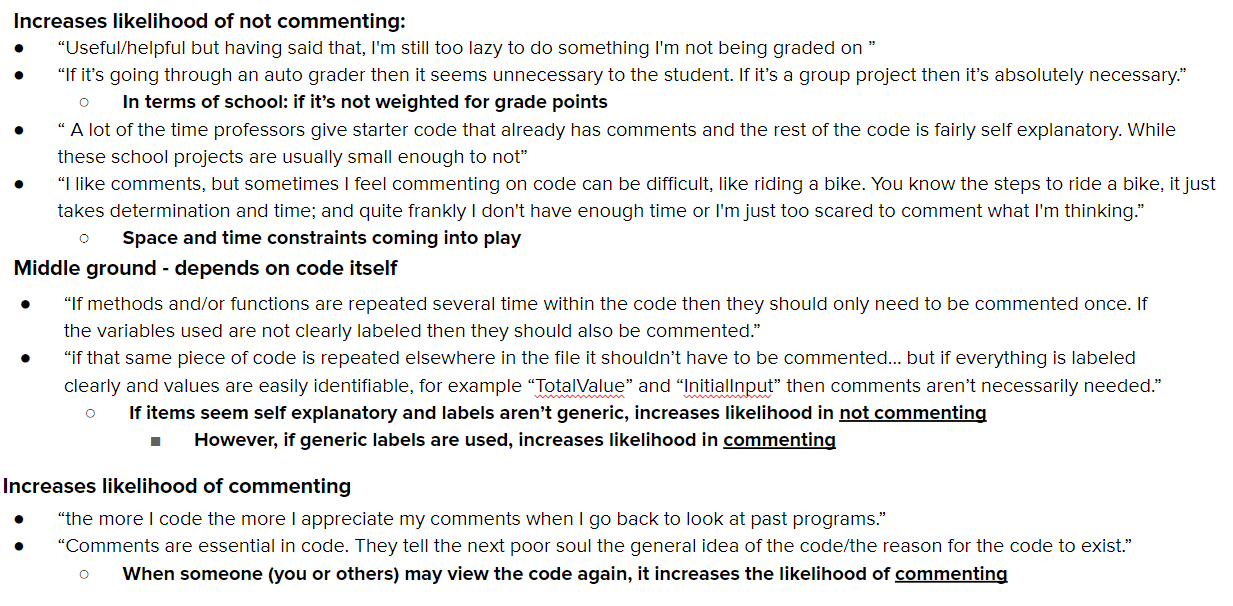
Too easy to put into words

* + - Determining if being optional was a factor
      * If it is optional, why would they go out of their way to do it
    - General comments and frequency of certain phrases
      * Mention notable comments, preferably the lengthier ones
* Since the main link to find and topic question is “Why don’t programmers comment their code” – Unit of Analysis is Programmers
  + Main topic shifted to relating factors of the programmer (external or internal) and commenting their code
* Instrument
  + Google forms sent out to subjects every 1-2 days for a period of a week
  + Consisted of few questions
    - 6, without counting “asking for 1st name”
  + As stated previously, meant to draw links as with various factors on why the programmer would comment or not comment

Sample:

* Sample Frame –would include all programmers
* Sample Size – 15 individuals
  + Max amount of time a person should have filled out form: 3 times
    - However, some did not fill it out all 3 possible times

Analysis:

* Graph 1 (Language):
  + most popular among participants were Python and Java, with C++ as a runner up
  + some single answers included C, HTML and Minecraft
* Graph 2 (Why the project?):
  + (48%) primarily program as part of their studies.
  + Next most frequent were those who program for fun (36%) and
  + The least of them actually have a job in the field (16%)
* Graph 3 (Supplied or from scratch?):
  + pie chart makes apparent that the majority of our subjects (64%) had been given code to work from, and that likely included comments
* Graph 4 (Reasons to not?):
  + 60% believe there are excuses not to document code
  + those who answered positively to this question typically seemed to believe that including comments was considered optional
* Graph 5 (Optional?):
  + overwhelming amount responded that they would, 72% of them
  + suggests that documentation is not completely lost on programmers and some do attempt to make an effort
* 

Recommendations:

* Maintain a much more effective reminding system for subjects
  + Some subjects had not filled out the survey the max amount of times possible with many doing it only twice
* Narrow the focus to an occupational one
  + For instance, focus on student programmers or industry programmers rather than programmers in general
* Change the question answers on some of the questions from simply yes/no

Conclusions:

* Responses from find that commenting code is important, but the failure to do so lies in many other factors
* If they are the only ones looking at the code, they find it pointless to do so,
  + But if it is to be graded or they are working in groups they will comment their code because either it’s required or to help others understand their code
* Many programmers have different styles of coding where others may not understand their code, so comments are a necessity
* Important to draw a line where spending too much time on commenting is a waste of time
* Even people who don’t program may be able to understand some snippets of code without comments or any prior knowledge of the code if it seems self-explanatory or is labeled well enough
* Commenting code walks a fine line between being too in-depth and being a waste of time, and being a necessity because without it there will be a loss of translation and confusion