**ST2 SS21 M0 Code Solutions Predictions – AI or Student?**

Please predict whether each code solution was generated by an AI chatbot, or written by a student.

So that the code is fully anonymised, each solution is stored in a single text file, where multiple Java files within a single solution have been concatenated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Solution name | AI or student? | In favour of student | In favour of AI | Confidence |
| 1 | student | * Borders are represented in a weird way (3-dim int tuple, origin + length) – I just don’t think an AI would do it that way * Use of “System.out.println” | * compact code * relatively clean | relatively strong |
| 2 | student | * The algorithm to split each command into a list of 1-point-move commands doesn’t sound “AI-like” * quite lengthy * maybe a bit “over-formalized”, trying to be a “good ST2 pupil …?” – corresponds with the somewhat over-fussy algorithm | * clean | weak |
| 3 | AI | * Sometimes inconsistent newlines directly after start of a method, or 2 newlines in a row | * very compact * a somewhat “no nonsense” style …, consistent and no unnecessary decorations * (really not much to use as criterium here …) | weak |
| 4 | student | * inconsistent spacing before / after “==” (sometimes with blank, sometimes without) * algorithm is a bit messy, a lot of methods and formalisms | * looks somewhat consistent; * uses IllegalArgumentException and IllegalStateException – I mean to have seen this before, that AI loves these exceptions * uses RegEx, also sth that AI likes | weak |
| 5 | student | * commented-out System.out.println statements (I guess both atypical for AI …?) * hardcoded obstacles into the for loops * relatively dumb code duplications for ea, no, … | * relatively short | weak |
| 6 | student | * the “barrier intersection” and the “if barrier in path” logic doesn’t look like AI, more like a from a human mind (a bit too “twisted” yet elegant) | * clean, relatively compact | medium |
| 7 | AI | * quite long | * looks clean and consistent | weak |
| 8 | AI | * …? | * very compact code * very uniform layout | strong |
| 9 | student | * uses Lombok * the Barrier enum implementation uses too many Booleans (one would have been sufficient – I don’t think AI would have done this) | * the JavaDoc comments – how many students do that?? |  |
| 10 | student | * Not very elegant code, no dedicated data structures / objects, instead using String or int arrays for tuples (see (1)) * inconsistent use of spaces after commas and declarations etc. | * very compact | relatively strong |