This document lays out your marks for the second coursework of CS141 Functional Programming.

Functionality

This solution goes significantly beyond the base specification to include a fully playable, interactive terminal UI for playing gridlock, based on a similar application produced by me for coursework 1. The imeplementation works exactly as expected, cleverly logging the "normal" game style within the window. Beyond this it also includes a working file browser for locating gridlock files to render. This implementation is excellent - very well done!

Elegance & Code Quality

The parser is split into a lexing step and an interpretation step, where it is first turned into an intermediate representation and then converted from that into a game. This will often result in a more lenient parser able to cope with varieties of presentation of games. There are some minor improvements to be made here; note the many copies of "space" in the parsers - this could be avoided by writing a function that runs a parser then consumes any trailing spaces (a common pattern is to call this a "lexeme"). Other simplifications would include changing doesPlayerExist to use "elem", and making more use of parser errors rather than bubbling storing move errors in the game record. Very well done for making use of the Brick library, which is famously tricky to get started with!

Justification & Documentation

The report is well written and beyond the simple explanation of what's been done, also includes video evidence of the program working. It would generally be preferable to write long-form text rather than solely bullet pointed lists, but it is still readable. The in-code comments are quite light and could have gone into more depth about some of the trickier aspects of the work, including the technical details around the use of brick.

Additional Comments

This is a superb program which shows excellent FP skill throughout and has been implemented solidly. Only some small improvements to reduce the complexity of the code, and explaining your work in more detail, could have improved the solution. Very well done!

Mark Breakdown

Part	**Mark**
Functionality	40
Elegance & Code Quality	28
Justification & Documentation	27
Penalties	-
Final Mark	**95 / 100**

Note: Any further adjustments (for example, lateness penalties) will be applied via Tabula and will not appear here.