# IB Computer Science IA Report

Battle Map Generator

Criterion E - Evaluation

# Criterion E: Evaluation

| Success Criterion | Met? | Evaluation and Future Improvements |
|---|---|---|
| Criteria 1 | Yes | Running a Python file is required at the moment. Could be a .exe file if run using other GUI libraries |
| Criteria 2 | Yes | It could include scalable windows. |
| Criteria 3 | Yes | Yes, could include more adjustable stats and an interface for changing default rendering colours for structures and cell types |
| Criteria 4 | Yes | Yes, as demonstrated in Criterion D. |
| Criteria 5 | Yes | Yes, although loading time scales with the number of maps linearly. Cannot handle upwards of 20 maps practically. However, a DnD campaign is unlikely to include more than 20 maps. |
| Criteria 6 | Yes | PysimpleGui, though provides a straightforward interface easy to build, restricts flexibility. This has led to me adapting to using several windows to act as panels for the map editor screen.<br><br>As for success criteria 6, although it is possible to add and delete cells and structures, the client finds it hard to operate(Appendix D). To combat this problem, I have already completed the GUI for placing the visualizer, however yet to be implemented. I still think it would've been better to implement it using a drag-and-drop system, though PySimpleGUI would be too limiting for that update.<br><br>Due to time constraints and limitations of the front end, many potential features are not implemented, such as the following:<br>- changing map sizes in the map editor<br>- combining maps in the map editor<br>- editing maps parallel to each other simultaneously in the map editor<br>- Selecting, copying and pasting map elements<br>- Interconnected multi-layered map |
| Criteria 7 | Yes | Yes, as demonstrated in criterion D. |
| Criteria 8 | Yes | Manually creating each cell type combination of tile texture would be arduous, whilst the TileSetGenerator class only has one aesthetically plain tile generation function. However, thanks to the flexibility of the cell_type_texture_heiarchy.csv, even with |

| | | |
|---|---|---|
| | | plain tiles, interesting aesthetics can emerge from these parameters which can be edited easily. |
| Criteria 9 | Yes | Yes. "...the panels is self-explanatory"(Appendix D) |
| Criteria 10 | Yes | The client mentioned a screen freeze during path generation(Appendix D). This is because it has a big O of O(xyn) where it is proportional to the number of tiles in the map's x and y axis and the number of structures. This path generation method would not scale well with size, however, common DnD games rarely need to use a battle map of this size and complexity.<br><br>The client also complains that the placement automatic structure generation is random and without many nuances(Appendix D), and stated they would use other commercially available map generation tools instead, despite their inconvenience. |
| Criteria 11 | Yes | Yes, as demonstrated in Criterion D, a map can be rendered from creation within 1.5 minutes. "I'll use it for westmarches" (Appendix D) |

Finally, the cell type CSV allows modders to easily add additional cell types, however, for structure types, other than manipulating the structure_types.csv file and adding the new structure subclass, they would also have to add instantiation parameters for functions that specifically identify the string names of the structure type, such as the structure list extraction in FileWriteHelper. As per the client's suggestion, there could be an interface for the implementation of a user-designed Structure / Cell type.