**- HTML SELECTION PAGE**

I need a dark themed (css) html input form that a user will use to select 2 things:

1) Tile sizes (from the list below). These are lengths and widths that are in feet

1 x 1

1 x 1.5

1 x 2

1.5 x 1.5

1.5 x 2

1.5 x 2.5

1.5 x 3

2 x 2

2 x 2.5

2 x 3

2) For each tile selected, a will select a quantity (from 1-10)

Submission of the form will pass the values to a python program.

After submission, a 'run tile' overlay will be presented showing the progress of the solution until vompletion…..at which point a results page is presented

**- PYTHON PROGRAM DUTIES**

1) Determine the 'optimum layout' of these tiles on a grid

2) Output the x&y coordinates, of the tiles from the generated layout, to a text file. The coordinates the program outputs need to be x&y of the left top most corner of each tile on the grid.

3) Program to create an html file that will display the layout of tiles on the grid. The display would show a 1px wide outline of each tile per the layout the algo generated

4) Program to create a run log showing the config values of the primary variables (MAX\_EDGE\_FT, PLUS\_TOGGLE, SAME\_SHAPE\_LIMIT) and entries for each phase below with run times, boards attempted and result

5) Program to create a 'Results' page upon completion displaying links to #2, #3 and #4

**- OPTIMUM LAYOUT**

The tiles will be linked to the grid. The grid will be comprised of 6" inch boxes. Laying out the tiles on the grid will follow these rules:

1) There can be no gaps between tiles in the layout at all.

2) The edges that form where tiles meet cannot be longer than 6 feet either vertically or horizontally (MAX\_EDGE\_FT=6 by default). So there will be no lines greater that 6' anywhere in the solution including the perimiter.

3) No four corners of four different tiles can share the same coordinate (PLUS\_TOGGLE=true by default)

for example, you can’t have four 1 x 1 tiles laid together in a 2 x 2 box orientation.

for example, all intersections must be an 'L' or a ‘T’ but and not a ‘+’.

4) The input must be distributed across the entire layout

No shape can share sides with 2 or more of the same shape ( SAME\_SHAPE\_LIMIT=1 by default)

**- SOLVER SEQUENCE**

The solver will iterate through the following phases:

If user provides a tile set LESS THAN 100 total sqf, then the solver should process these 2 phases in sequential order:

PHASE A. Fit all tiles to a SQUARE grid that is the smallest L&W that can be fully covered. Find start\_leg which is sqrt of the total\_sqf rounded up to nearest foot.

Starting grid length and width will be start\_leg + 2 feet X start\_leg + 2 feet.

Provide the solution or try for a maxumum of 10 mins (PHASE\_A\_TOT=600sec by default) reducing the square grid boards as necessary to get a full coverage solution….if not then go to next phase

PHASE B. Fit all tiles to a SQUARE grid that is the smallest L&W that can be fully covered . Find start\_leg which is sqrt of the total\_sqf rounded up to nearest foot.

Starting grid length and width will be start\_leg + 2 feet X start\_leg + 2 feet.

Allow pop-outs with mirroring AND/OR discarding if it makes any board solvable

Provide the solution or try for a maxumum of 10 mins (PHASE\_B\_TOT=600sec by default) reducing the square grid boards as necessary to get a full coverage solution….if not then exit to results page

"….If either option works, message the user that this is the best that can be done

...If there are no solutions reached within these timeframes then message ""there is no solution within the provided timeframes"""

If user provides a tile set EQUAL TO OR MORE THAN 100 sqf then the solver should process these 4 phases in sequential order:

PHASE C. Fit all tiles to a SQUARE grid that is the smallest L&W that can be fully covered. Find start\_leg which is sqrt of the total\_sqf rounded up to nearest foot.

Starting grid length and width will be start\_leg + 2 feet X start\_leg + 2 feet.

Provide the solution or try for a maxumum of 10 mins (PHASE\_C\_TOT=600sec by default) reducing the square grid boards as necessary to get a full coverage solution….if not then go to next phase

PHASE D. Fit all tiles to a SQUARE grid that is the smallest L&W that can be fully covered. Find start\_leg which is sqrt of the total\_sqf rounded up to nearest foot.

Starting grid length and width will be start\_leg + 2 feet X start\_leg + 2 feet.

Allow pop-outs with mirroring AND/OR discarding if it makes any board solvable

Provide the solution or try for a maxumum of 10 mins (PHASE\_D\_TOT=600sec by default) reducing the square grid boards as necessary to get a full coverage solution….if not then exit to results page

"….If either option works, message the user that this is the best that can be done

...If there are no solutions reached within these timeframes then message ""there is no solution within the provided timeframes"""

- STRAWMAN PROJECT STRUCTURE

app.py

config.py

logs

>>>run\_log

templates

>>>tile\_selection\_form.html

>>>results\_form.html

>>>styles.css

outputs

>>>coords.txt

>>>layout.html

solver

>>>orchestrator.py

>>>other solver helper files