More details on the “Latin Square Type of Error”.

(1) Notice that the test for menus in our book (computer pull down menu example) uses MS(menu*square) as its denominator (pg. 129). I mentioned this in class. The menu*square degrees of freedom number is (t-1)(s-1) when you have s squares and t treatments (menus). In other words the main test of interest is unaffected by whether we pool or not in that table, but see also points 2, 3 and 5 below.

(2) In our example, rows, columns, and treatments (i.e. menus) are really fixed factors that are assumed not to interact with anything, so you can interpret that (12 df) three way interaction as measuring a combination of interactions among those three sources, that is, a check on our assumptions. If you put in row*column, row*menu, column*menu and row*column*menu in some order, whatever gets in first will give you those 12 df and will get all of that Type I sum of squares that we specified as row*column*menu in class. Of course Type III will be 0 for all of those effects since Type III is what you’d get if that effect were fitted last. In other words a lack of estimability has rendered us unable to separate out the potential interactions from each other!

If rows and columns were random you could interpret that 3-way interaction more or less the same way but thinking now of the row*column part as measuring lack of independence. This may help in interpreting what is called “Latin Square Type of Error” in our book. As we saw in class if we declare this three way interaction (Latin Square type of Error) to be random (i.e. thinking of it as really an error term) then in contradiction to point (1) we get a mixture of mean squares as our menu F test denominator, that is, it is important to correctly classify the rows and columns as fixed or random. The comment on page 129 of the book is assuming (correctly) fixed rows and columns (monitors and lighting).

(3) We would really hope that the multiple squares are acting as blocks, meaning that all interactions with square can be pooled with the MSE. That is where the real gains in running multiple squares would occur as you would then boost your degrees of freedom considerably over a one square experiment. You would have st^2-1 df total and loose only t-1 for each of row, column and treatment and you’d also loose s-1 df for squares (blocks) so that you’d have st^2-1 -3(t-1) – (s-1) error df. That can be a lot more than (t-1)(s-1).

(4) Degrees of freedom.
With s squares and t treatments we have t rows and t columns as well. In our computer example, the rows and columns represent the same thing (monitor design, lighting) across the squares and hence would most likely be considered fixed. The degrees of freedom as well as proper F tests depend on what sources are in the model. Here is the most complete breakdown of sources and df:

Table 1
Squares s-1
Rows t-1
  Rows*Squares (s-1)(t-1)
Columns t-1
  Columns*squares (s-1)(t-1)
Trts t-1
  Trts*squares (s-1)(t-1)
Row*Col*Trt= Latin Square type error (t-1)(t-2)
Residual Error (s*t*t-1)-(s-1)-3(t-1)-3(s-1)(t-1)-(t-1)(t-2)= (s-1)(t-1)(t-2)

We would hope to pool the Latin Square type error with the remaining error (which can also be thought of as Latin square type of error x square- see Table 6.6)   In this case we have

Table 2

Squares	s-1
Rows	t-1
Rows*Squares	(s-1)(t-1)
Columns	t-1
Columns*squares	(s-1)(t-1)
Trts	t-1
Trts*squares	(s-1)(t-1)
Error	s(t-1)(t-2)

(5) Tests:
Assuming squares are random, rows, columns and treatments in Table 2 are tested against their interaction with square. If the model contains more terms (like the Latin square type of error as in Table 1) and that "Latin square type of error" is considered random then, as we saw in class, it will take a mixture of mean squares to construct a proper denominator, and we'd have to use Satterthwaite's method for degrees of freedom.   It seems pretty clear to me that the monitors and lighting would be fixed if we really did this experiment.

We might hope to pool further, for example removing some of the interactions with square so they get pooled with our error term. In this case those degrees of freedom (s-1)(t-1) would be added into error degrees of freedom as well.   The menu*square term can definitely not be pooled with error in the class example and would be used in Table 2 as the proper error term for menu.

Finally note that each square has not been rerandomized so the same menu, light, and monitor combinations are used in every square. That is the only way that row*column*menu even has a meaning so the breakdown in Table 6.6 (Table 1 above) would not even be considered were this not the case.