DØG
A DECISION MATHEMATICS GAME

Geoffrey Churchill, Georgia State University

INTRODUCTION

DØG [1;2;3] is a small scale deterministic business game designed to provide an environment in which to teach elementary modeling and Decision Science concepts. Most of the assignments utilize matrix algebra as the fundamental tool, although others do rely on calculus, linear programming, or free form modeling. Little or no prior background in business, economics, or accounting is required.

Each quarterly decision in the game is associated with a specific assignment, although the assignment will not relate to all items in the decision. It is not necessary to use all assignments, but some assignments do rely on previous assignments. For this reason it is highly desirable to plan quite carefully prior to selecting a sequence of assignments to use. It is normally quite feasible to assemble a sequence which will be consonant with both available tine and course objectives. Needless to say, before a final decision is made on the assignment sequence, the assignments tentatively chosen, as well as their solutions, should be read.

The amount of classroom time devoted to the game is optional, depending upon course content and objectives and upon the level of student preparation. A rule of thumb that is descriptive of its use at Georgia State University is that about 25% of the class time is directly game-related; some instructors have also designed game-related supplementary materials and exercises to take advantage of heightened student interest and of familiarity with the simulated environment.

The simulated environment was originally written is BASIC, to take advantage of a large time-sharing system with a number of convenient features. Since these convenient features are not universally available, a Fortran IV version has since been developed.

The BASIC version utilizes two BASIC programs. PUPPY is a small program which initializes files prior to the start of play. DØG is the game program. As written, DØG requires a time-shared BASIC system with the following features:

Remote terminal Input/output
Line printer access through LIST statements
Capacity to access two disk or drum data files
Capacity for a 949 statement program
Capacity for 279 dimensioned storage locations

While certain of these requirements can be evaded through modification of the program, such modifications carry a significant penalty in making the game awkward to use.

Whereas the BASIC version of DØG requires a time-sharing computer system, the Fortran version is hatch-oriented. The structure of the two programs in the Fortran version of the game is somewhat different than that of the BASIC version. In the BASIC version, PUPPY is run once to initialize a history file, and thereafter only the program DØG is used. The Fortran version also uses PIJPPYP to handle the input data for each run: thus a run of PUPPYF will precede each run of DØGF. Otherwise, the Fortran algorithm is simply a translation of the BASIC algorithm. There are no differences in logic.

The major system requirements for the Fortran IV version are:

Card reader input
Line printer output
Capacity to access three disk or drum data files (or, with more difficulty, tapes or cards could he used)
Capacity for a 559 statement program
Capacity for 538 dimensioned storage locations
Capacity for 66K Bytes of object code (This will vary somewhat by system.)

In addition to the game programs, several small BASIC programs are provided. In all cases they run successfully on a WANG 3300, one of the smaller general purpose mini-computers available. They are designed in each case to reduce the student's computational effort, not to replace thoughtful modeling. They are not essential to the successful use of the game hut are a useful adjunct where it is feasible to make them available for student use.

STRUCTURES

This, like most games, is based upon an interlocking set of structures. An understanding of the total game environment is best based on an understanding of the more fundamental structures, which are therefor outlined in this section.

Decision Structure

Imbedded in the model is a set of default decisions/decision rules, as follows:

Price: Fixed.

Advertising: Fixed.

Subassembly production: Enough to meet. production requirements and leave 100 unit ending inventory of each subassembly.

Product production: Enough to meet demand plus requirements For use as a subassembly and leave 100 unit ending inventory of each product.

Materials purchases: Ten equal orders sufficient to meet requirements and leave approximately a 100 unit ending inventory each material.

Labor Force: Enough men to meet requested production (regardless of feasibility).

Borrowing/repayment: None.

Dividends: Five hundred dollars.
Equipment purchases: None.

Any of these default decisions can be over-ridden by student decision; the selection of assignments determines the pace at which students replace these automatic rules. Any totally infeasible decision is automatically modified into a Feasible, but usually quite poor, decision.

Demand Structure

Let: T = quarter number
A = Advertising, by product
$\bar{A}$ = Industry average advertising, by product
P = Price, by product
p = Industry average price, by product

The structure is:

Early play (no price, advertising decisions):

$$\text{Demand (1)} = \hat{a}_1 + b_1T$$
$$\text{Demand (2)} = \hat{a}_2 + b_7T + c_2T^2$$
$$\text{Demand (3)} = \hat{a}_3 + b_3T + c_3T^2 + d_3T^3$$

Later play (with price, advertising decisions):

$$\text{Average Demand (1)} = a_1 + b_1T + c_1\, p + d1\, \bar{A}$$
$$\text{Average Demand (2)} = a_2 + b_2T + c_2T^2 + d_2\, (\bar{A}/p)$$
$$\text{Average Demand (3)} = a_3 + b_3T \div c_3T^2 + d3T3 + e3\, (\bar{A}/p)$$

Demand (product, firm) $(A/\bar{A})$ (p/P) Average Demand (product)

Note that neither here nor elsewhere is any random component present.

Production Structure

The essence of this structure is best described as a sequence:

Production decision (8 items)
then
Find most constraining process (3 processes)
then
Trim all items equally, if constrained
then
Find labor hours available
then
Trim all items equally, if constrained
then
Find most constraining raw material

then
Trim all items equally, if constrained
then
Going from least complex item to most complex item decrement inventory of required subassemblies, trim output if constrained, increment inventory of resulting subassemblies and products.

All usages (capacity, labor, materials, subassemblies) are constant at all levels and known to the student.

Financial Structure

A direct costing system is used; no allocations of indirect costs to products are attempted. Statements provided include a Cost of Goods and Inventory Statement in an input-output format, a Cash Flow Statement, an Income Statement, a Balance Sheet, and an Industry Report. Cash deficits are not permitted; high cost emergency loans are automatically issued when needed.

ASSIGNMENTS

The structures on which the environment is based were, in turn, developed to fit the set of assignments which are designed to integrate the game into a college entry level mathematics course with a business decision making orientation. Somewhere in the set of objectives motivating an instructor to bring a game into a course there must be some rather specific learning goals. This game is designed to permit a rational selection of such goals in a mathematics context, as may he seen from the assignment list[1]:

Assignment	Topic (methodology)
1	Graph and project demand (graphing)
2	Total materials orders (matrix multiplication model)
3	Capacity constraint and employment (matrix multiplication model)
4	Forecast average demand, product 1 (linear curve fit)
S	Forecast average demand, product 2 (quadratic curve fit)
6	Forecast average demand, product 3 (cubic curve fit)
7	Forecast capacity utilization (matrix multiplication model)
8	Make or buy subassembly 3 (breakeven model)
9	Industry price and advertising effect, product 1 (multivariate curve pit)
10	Industry price and advertising effect, product 2 (multivariate curve fit)

[1] From [2], p. 7.

11	Industry price and advertising effect, product 3 (multivariate curve fit)
12	Optimize product mix (linear programming)
13	Evaluate equipment purchase (linear programming sensitivity analysis)
14	Production scheduling (parts requirement model)
15	Make or buy subassembly 3 (cubic breakeven model: calculus of extrema)
16	Raw material inventory costs (general modeling; curve fitting)
17	Economic order quantity (minimization of hyperbolic model)
18	Share of market and exact demand (general modeling)
19	Revenue maximization, product 1 (maximize quadratic function)
20	Revenue maximization, product 2 (maximize hyperbolic function)
21	Revenue maximization, product 3 (maximize cubic function)
22	Profit forecasting (least squares fit of exponential function)
23	Wealth forecasting (definite integral of exponential function)
24	Product mix risk balancing (critical ratio expected marginal value model)
25	Accounting forecasting (general modeling)

While partial precedence requirements do exist (e.g. assignment nine requires prior completion of four and any of six, seven, or eight), it is not necessary to march through the assignments in strict numerical order. This facility is essential; the game must not be permitted to dominate the order of topics in such a course.

A SYNTHESIS OF EXPERIENCE

This game is now in its fourth quarter of use, having been through two major revisions of both the manuals and the programs. While it is always difficult to “prove” the worth of any educational tool, certain points have surfaced either so repetitively or so emphatically that they cannot be ignored. These points seem to divide quite naturally into three distinct categories.

Misuse

It is clear that DØG is both a delicate and a powerful instrument, and as such is both easy and dangerous to misuse. It is tempting to simply overlay an existing course outline with the game; this certainly holds down the preparation effort required of the instructor. It is similarly tempting to require two or three decisions and/or assignments per week, perhaps in an attempt to exhaust the list of relevant assignments. Unfortunately any of these courses of action will also exhaust the students, thus creating resentment toward course, game and instructor, and possibly distracting students from the fundamental elements of the course.

Another way to cause the game to bomb also reduces the instructor's preparation time. It is really quite simple: don't bother to learn the environment, don't read the manuals, don't explore the assignments, and above all, never attempt to play the game yourself. Students seem much more comfortable when they can regard their instructor as an expert; destroy the image of expertise and they may assume that expertise is too difficult to attain. Not learning the environment also prevents the instructor from editing student decisions to prevent major disasters.

A final way to generate difficulty is also tempting. Place heavy emphasis on firm performance; really put the students under pressure. This results in very conservative play, a heavy game workload, and intense frustration when an early misjudgment puts a firm far behind.

Successful Use

One component of successful use is, of course, to avoid the misuses cited above. Beyond that, the most critical thing is a matter of attitude. If the game is seen as a pleasant vehicle and context for learning mathematics, then the experience will be a good one. One implication of this is that a student's successes in running his firm should be reinforced, while his failures are treated as something from which to learn.

It seems strange to most students to need to search their operating results to find the data relevant to an assignment. The most successful instructors using the game have evolved a pattern or "style" that eases this transition. First, the methodology to be used, the fundamental course material, is presented, prior to student exposure to the assignment. The assignment is introduced by the instructor, not merely announced. The extent of introduction will vary by class, since not all students require the same amount of guidance. Some assignments should even be done by the instructor at the board or on handouts. All $_0$F this is in line with the notion that the game is pot the course; the purpose of each assignment is to provide a "live" and interesting application of one or more concepts from the course.

Benefits

When properly used, DØG does seem to consistently provide useful benefits. Perhaps the most crucial of these is student involvement in the course. This is almost inevitably cited as an advantage of gaming, but it is perhaps more critical here than in many courses. A large and vocal subset of undergraduate business students seem to regard mathematics courses as a collossal and irrelevant imposition. For many of these, the DØG experience erodes their objections to the point where quite a few even become enthusiastic about the course. Some even get so involved that they actually study even when no quiz has been announced.

Beyond the question of involvement, there are other benefits to be obtained. A typical goal in a business-oriented mathematics course is the development of situational problem-solving expertise, and toward that end large numbers o "word problems" are frequently used. DØG cannot entirely replace word problems, but it can enhance them. Beginning students often have little or no business/economics background; developing a bill of materials by matrix multiplication or maximizing a revenue function through the calculus of extrema frequently presents no clearer an issue than if they were asked to vorple the gort with a turboencabulator. The game forces questions of terminology to the surface early in the course. Furthermore, continuing use of an environment that all students then have in common provides a universe of discourse accessible to all. Since half the difficulty in situational problem-solving is in grasping the situation, this common universe of discourse can provide a bridge toward development of general problem-solving capabilities.

REFERENCES

1. Churchill, Geoffrey and Rachel Elliott Churchill, Decision mathematics Operational Game, (Atlanta), Georgia State University, 1974.

2. Churchill, Geoffrey, Decision mathematics Øperational flame Instructor's Guide, (Atlanta), Georgia State University, 1974.

3. Churchill, Geoffrey, FØRTRAN IV Supplement to Decision mathematics Øperational Game Instructor's Guide, (Atlanta), Georgia State University, 1974.