ANOTHER CHANCE: A MULTI-LEVELED, MACRO BUSINESS GAME FOR MICROCOMPUTERS

Precha Thavikulwat, California State University, Los Angeles

ABSTRACT

ANOTHER CHANCE is a microcomputer business simulation game whose distinguishing features are its progressive difficulty, its macro scope, and its Monte-Carlo, independent-across-players, design. The game is meant to be used as a homework assignment, with no handling required of the instructor. Business concepts encompassed by the game include sales forecasting; elasticity of demand; inventory control; fixed and variable costs; lease versus buy; cash flow; income and expenses; and assets, liabilities, and equities.

INTRODUCTION

For some time now, the educational benefits that can be gained from the use of microcomputers to support educational games students could run on their own time and at their own pace has been evident. Yet, despite the prevalence of microcomputers today, few college-level games for microcomputers are available. Although in business schools, computer games that are macro in scope, where students make fundamental decisions and see the results in financial statements, have been used for over two decades, these games were designed to run on mainframe computer systems. The mainframe games could be transported to microcomputers, and in fact, this has been done with several games [2, 61.

But to take full advantage of the microcomputers' unique attributes and account for its particular limitations, a game should really be specifically designed for microcomputers. AIRWAYS by Fisk, Gentry, and Fisk [4] is one such game. AIRWAYS, however, is made for team play in the spirit of the older mainframe games. This paper describes ANOTHER CHANCE, a simulation game also specifically designed for microcomputers, but different from AIRWAYS in that ANOTHER CHANCE is made for individual play.

For a microcomputer game designed for individual play, the choice of a programming language is critical. Because of the great variety in makes and models of microcomputers, the language used should be highly standardized for easy transport among microcomputers of different manufactures. Because the entering of data will necessarily be done by inexperienced students, the language must be flexible and powerful enough to allow the program to maintain complete control of the data-entering process. And because slow computational speed is most irritating when working on a microcomputer, the language must be close to machine language so as to permit compilation to fast-running executable codes.

The C language, as defined by Kernighan and Ritchie [7], meets these requirements well. The C language is currently the standard professional language of microcomputer programming. ANOTHER CHANCE was written in C, and compiled with the DeSmet C compiler [3], a compiler noted in a review by Phraner [8] for generating code that runs fast, as well as for being an excellent overall value. The game was written on a Sanyo MBC-555. It will run on IBM PC's and compatibles under MS-DOS 1.25 or MS-DOS 2.11. And because C is a highly standardized language, the code can be compiled, with an appropriate C compiler, for any machine.

ANOTHER CHANCE is designed to be used as a homework assignment, with no handling required of the instructor. Data from running the game is stored on the program diskette and encrypted so as to be unreadable by direct display, thus preventing students from cheating by directly altering the data file. The instructor obtains evidence of progress with assignment by requiring the students to submit printouts from time to time.

The distinguishing features of the game are that it is multi-leveled, or progressive in difficulty; it is macro in scope, that is, results are displayed in financial statements; and it is of a Monte-Carlo, independent-across-players design. As a multi-leveled game, the difficulty of the game increases, and the number of decisions students are allowed to make also increases, as targets are reached. In this respect, ANOTHER CHANCE follows the principle of progressive difficulty seen in popular video games. As a business game macro in scope, ANOTHER CHANCE tests students' understanding of the cause and effect relationship between operating and marketing decisions on one hand, and financial results on the other. And as a game of Monte-Carlo, independent-across-players, design, a design with advantages discussed by Chiesl [1], ANOTHER CHANCE is self-paced: each student can run the game through as many periods as the student finds necessary to reach each profit target.

It is in its self-paced aspect that ANOTHER CHANCE takes advantage of the independent processing afforded by microcomputers, as different from the central processing of mainframes. With conventional mainframe games, because of administrative considerations, the number of periods of play is fixed. But with ANOTHER CHANCE, the number of periods of play is unlimited: the student has another chance, and another, and another . . . without end. Thus, although the student may eventually give up on the game, the game never gives up on the student.

LEARNING OBJECTIVE

Games confront students with new concepts, and make them more comfortable with those they already know. Students playing ANOTHER CHANCE will have occasion to apply the following concepts:

1. forecasting sales
2. elasticity of demand
3. inventory control
4. fixed and variable costs
5. lease versus buy
6. cash flow
7. income and expenses
10. assets, liabilities, and equities

GAME DESIGN

The game has three modes: identification mode, entry mode, and display mode. When the game is run, it first enters identification mode, where the student is prompted for name and password. Given the correct identification, the game moves into entry mode, where the student is prompted to enter decisions for the run in full-screen entry format. From entry mode, the student may, at the student's option, move into display mode, where the student can view displays of the facts of the game. Decisions are executed within display mode, and the results may be seen immediately.

The game provides five displays: the decisions display, the parameters display, the operations display, the income statement display, and the balance sheet display. The student may view the displays, two at a time in any combination, in side-by-side display windows. A print-screen and a print-all feature for hard copies is included.

The decisions display doubles as the full-screen form for data entries in entry mode. It shows the decisions made for the run. These decisions are as follows:

1. units of production
2. product price
3. units of material to purchase
4. units of building to purchase
5. units of equipment to purchase
6. units of equipment to lease

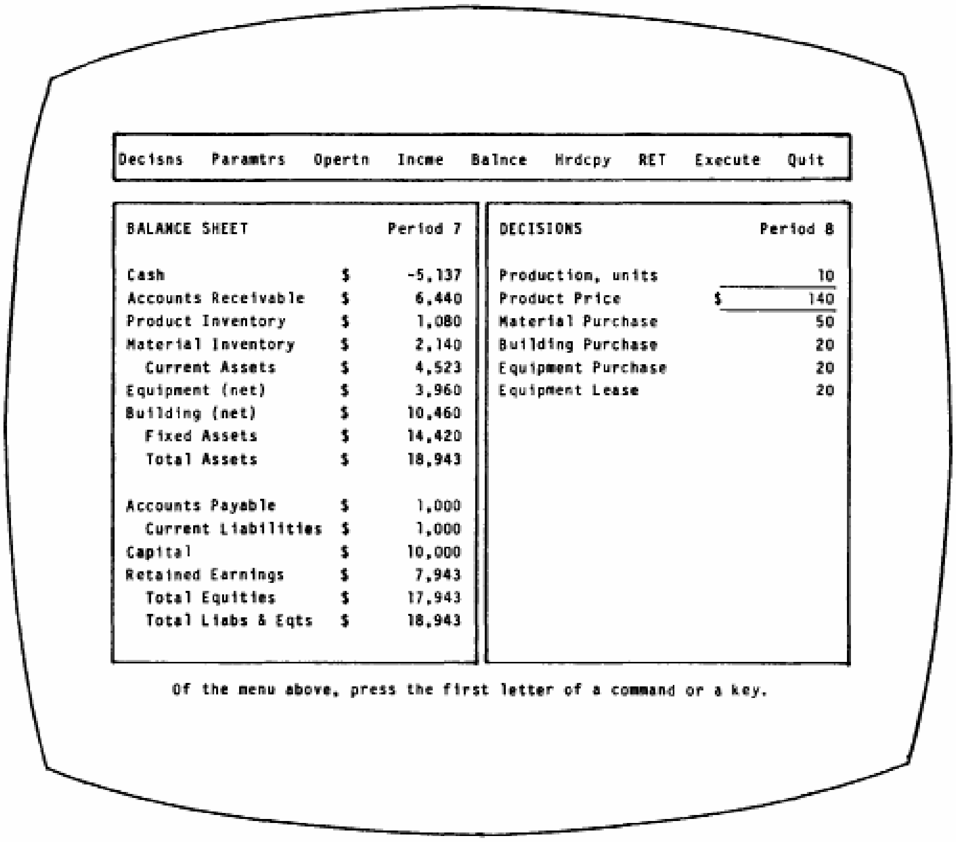
The parameters display shows the parameters of the run. These are as follows:

1. earnings target
2. target salient time
3. level of difficulty
4. number of seasons
5. demand ceiling
6. building depreciation rate
7. equipment depreciation rate
8. interest rate
9. ordering cost
10. unit material cost
11. unit labor cost
12. unit labor cost
13. unit building cost
14. unit equipment cost
15. unit lease cost

The operations display shows the operating facts, which are as follows:

1. product demand
2. beginning product inventory
3. production
4. products available
5. products sold
6. ending product inventory
7. beginning material inventory
8. material purchase
9. materials available
10. materials used
11. ending materials inventory
12. beginning building capacity
13. ending buildings capacity
14. beginning equipment capacity
15. ending equipment capacity

FIGURE 1
EXAMPLE OF A SCREEN OF DISPLAYS



```
BALANCE SHEET              Period 7    DECISIONS                  Period 8

Cash                   $     -5,137    Production, units                10
Accounts Receivable    $      6,440    Product Price           $       140
Product Inventory      $      1,080    Material Purchase                50
Material Inventory     $      2,140    Building Purchase                20
  Current Assets       $      4,523    Equipment Purchase               20
Equipment (net)        $      3,960    Equipment Lease                  20
Building (net)         $     10,460
  Fixed Assets         $     14,420
  Total Assets         $     18,943

Accounts Payable       $      1,000
  Current Liabilities  $      1,000
Capital                $     10,000
Retained Earnings      $      7,943
  Total Equities       $     17,943
  Total Liabs & Eqts   $     18,943

Of the menu above, press the first letter of a command or a key.
```

The income statement display shows the following:

1. revenue
2. material cost
3. labor cost
4. cost of goods sold
5. gross margin
6. ordering expense
7. lease expense
8. building depreciation expense
9. equipment depreciation expense
10. total operating expenses
11. operating income
12. interest income
13. interest expense
14. net income

And the balance sheet display shows the following:

1. cash
2. accounts receivable
3. product inventory
4. material inventory
5. current assets
6. equipment (net)
7. building (net)
8. fixed assets
9. total assets
10. accounts payable
11. current liabilities
12. capital
13. retained earnings
14. total equities
15. total liabilities and equities

An example of a screen of displays is given in Figure 1. The top box of the screen is the menu window, where available options are shown. The pair of boxes below it are the windows for the displays. Helpful messages in reverse video appear in the one-line space directly under the display windows.

The game is progressive in difficulty, and progressive in degree of control. After a minimum number of periods have elapsed at each level of difficulty, the earnings target becomes salient. Subsequently, when cumulative earnings reaches the earnings target at that level, the game progresses to the next higher level, which has a new target, a more difficult set of parameters, and a greater number of allowable decisions.

Sales demand is bounded by a ceiling and a floor. It follows a linear relationship with respect to time such that it “bounces off” the ceiling and the floor. It varies randomly, and the random component is normally-distributed. It is also seasonal, with the number of seasons increasing as the game advances in difficulty.

The parameters of the game are preset to change as the game moves from one level to the next. These settings may be altered by changing the headers of the program codings, and by recompiling the codes. The procedure is not difficult, but does require a rudimentary knowledge of the C programming language, and access to a C compiler. A utility program to change parameters, as suggested by Fritzsche and Cotter [51, will be made available at a later date. Alternatively, the parameters may also be changed by directly modifying the compiled program using widely-available debugging utility programs.

CONCLUSION

ANOTHER CHANCE is a simulation game specifically designed for microcomputers. It is a game that is progressive in difficulty and macro in scope. It is a game on which students may work at their own pace, a game that the instructor can assign and almost forget. The game takes care of itself, and is infinitely patient with students.

The game was written in the C programming language. The power of this language enabled the game to have the characteristics of speed, of controlled input, and of clean display especially desirable in microcomputer games of any kind, whether educational or frivolous. Perhaps other educators will also find C to be the language of choice for microcomputer games of this age.

REFERENCES

[1] Chiesl, Newell, “A Monte-Carlo Approach to Interactive Gaming,” Developments in Business Simulation & Experiential Exercises, Vol. 12, 1985, pp. 78-81.

[2] Decker, Ron and James LaBarre, “Bus-Sim Micro: An Introductory Business Simulation Demonstration,” Developments in Business Simulation & Experiential Exercises, Vol. 12, 1985, pp. 22.

[3] DeSmet Software, DeSmet C Development Package Manual, (Sunnyvale: C Ware Corporation, 1984).

[4] Fisk, Jamie T., James W. Gentry, and Raymond P. Fisk, “Airways: A Microcomputer Simulation of a Service Industry,” Developments in Business Simulation & Experiential Exercises, Vol. 12, 1985, pp. 127-130.

[5] Fritzsche, David J. and Richard V. Cotter, “A Guide to Writing Microcomputer Simulations”, Developments in Business Simulation & Experiential Exercises, Vol. 12, 1985, pp. 69-73.

[6] Jensen, Ronald L., “The Business Management Laboratory on a Microcomputer,” School of Business, Emory University.

[7] Kernighan, Brian W. and Dennis M. Ritchie, The C Programming Language, (Englewood Cliffs: Prentice-Hall, Inc., 1978).

[8] Phraner, Ralph A., “Nine C Compilers for the IBM PC,” Byte: The Small Systems Journal, Vol. 8, No. 8, August 1983, pp. 134-168.