# MICROCOMPUTERS IN WARGAMING

MIKE COSTELLO

Microcomputers are in the news lately; they are spreading into an increasing number of houses as the price falls drastically. Some idea of what you can get for your money can be gained by looking at the 'Budget' panel with this article. Brief references to the role of microcomputers in wargaming have appeared in wargaming magazines recently, but there has been no sustained examination, and the attitude of the average wargamer is probably one of faint interest tinged with mild distrust. During 1979 a number of references appeared in **S&T** and **Moves** to **SPI**'s plans to launch a Newsletter **(SpiBus)** for wargamers with computers, together with a range of game-assistance programs running on the **TRS-80** microcomputer. All these projects were subsequently quietly dropped, and it would be easy to get the impression that, after an initial surge of enthusiasm, game designers have found computers to be intrinsically unsuitable for wargaming.

Nothing, however, could be further from the truth. The fact is that these machines offer such striking advantages to wargame designers and players that they can't be ignored; they will shortly begin to play a central role in wargaming activities and it would not be a bad idea to take a look at what they can and can't offer.

### What the computer can do .....

The first attempts to visualise computers in wargaming led to the idea that the TV screen (VDU) of the computer would replace the traditional hexagon mapsheet, and the game would be played out on the screen, the computer calculating the movements of the units. Although attractive, this idea was impractical, not so much because of any limitation in the capabilities of microprocessor units nowadays, but rather because the technology of "peripherals" has lagged behind the development of the computing systems themselves. A peripheral is any kind of device which is linked to the computer in order to push data into it (like a keyboard) or to display data from the computer in a comprehensible form, as in the case of a VDU, which displays the results of the calculations performed by the computer on a TV screen together with an indication that the computer is ready for the next job. Although VDU's are fine for displaying text and numerals, they do not make a very good job of representing a typical wargame map, and they are after all only cathode ray tubes – small, bulky and fragile. One day we will have flat-screen TV, but until then there is no possibility that the standard hexagon map and countersheet of the typical wargame will be superseded.

A more productive approach is to start by considering what computers are good at, and then see how these special capabilities of computers can be turned to advantage. Computers can offer help in at least four areas of wargaming activity, and what makes them so valuable is that these are areas which have presented designers and players with particularly intractable problems during the last ten years or so.

In a moment we will go on to look at these four areas in more detail. It should be borne in mind, however, that there are two different applications for computers in wargaming, and these have to be treated separately. The first application can be described as the game-assistance program, in which the computer can, for example, replace the charts and tables used in a typical wargame, and give the players the results of combat resolution procedures and other routines without the need for tedious calculations by the human participants. The second application lies in the realm of Artificial Intelligence, since it concerns the replacement of one of the human players of the game by a computer program which simulates the intelligent actions of a player. The first application is simply waiting to be exploited, since there are no theoretical obstacles; it has not been commercially attractive until now, since the owners of domestic microcomputers who are also keen wargamers form a very small section of the market. This situation is now changing rapidly, partly because of the sharp drop in prices of micros over the past year and partly because of the advent of very cheap machines like the **ZX-80**, which, although it hardly has the capacity for complicated programs, is going to whet the appetites of a lot of people. The second application is not quite with us yet as it involves writing some new types of program which will be a considerable extension of current work in Artificial Intelligence, but in the meantime we have one or two partial solutions like **"Tanktics,"** of which more later.

### Simulating Limited Intelligence

All game designers agree that some aspects of combat are more difficult to simulate than others and the most intractable problem has always been in an effective simulation of the lack of information available to commanders of the opposing sides in the majority of situations. In fact this has probably distorted the whole history of wargaming since designers have tended to steer clear of campaigns and battles in which Limited Intelligence was a decisive factor. A number of different sets of mechanics have been devised, each with restricted application. A good example of the problems faced by players and designers is the debate in recent issues of **Moves** concerning double-blind naval searches.

For those of you who are not familiar with this magazine or this problem, the trouble arises whenever a designer wants to simulate the reconnaissance phase leading up to a battle between opposing naval fleets containing aircraft carriers. The American player (say) wants to search specific hexes using air units on search patrol; he has to be told if any of the hexes searched contains a Japanese unit. But if he simply tells the Japanese player which hexes he is searching this will reveal the approximate location of his own fleet, since aircraft have a comparatively short range. He wants to be able to search a hex, establish that it is empty, and return to base without revealing his location. On top of that, it should sometimes be possible for an aircraft to spot a surface fleet and return to the carrier without revealing itself to the enemy – and *that* is a double-blind search system, if it can be achieved.

In **Moves 36,** David Clark suggested a system for coding hexes on a wargame map such that each player would not be giving his position away. Without going into details, it turned out that the system was unworkable and a letter from Friedrich Helfferich in **Moves 37** pointed out that no such system could be devised. Dieter Schlaepfer, in

Moves **39**, **came up with an alternative system** which turned out to be too unwieldy to use; V J Thomas in **Moves 40** came forward with an apparently foolproof plan which took two pages of equations and charts to expound. Finally, in **Moves 44**, Rudolph F Lauer showed that all the previous systems, including Thomas', wouldn't work, and described another one. It involved revealing the exact number of hexes occupied by one's forces each turn, and didn't seem to offer any more security than the others; the assumption in all these articles was that both players are mental geniuses, continually attempting to break each other's codes in order to "beat the system."

Nothing more has been heard on this subject in **Moves**, and in retrospect, it would have been better to take Fred Helfferich's advice. In most games, a simple concealment system will give the players some inkling of the problems of limited intelligence, and any more complex system will be too unwieldy and still not totally secure. The series of articles in **Moves** illustrated vividly the severe limitations of the conventional board wargame; there are several aspects of warfare it simply can't cope with. The authors of the articles in **Moves** ended up by recommending that the players purchase about £250 worth of calculators to encode hexes, and yet this money would have been much better spent on £100 worth of computer keyboard, to be plugged into the TV and used to resolve all the searches with no human effort at all. The sample program in the "double-blind search" panel (p.10) shows one way of doing this (not necessarily the best) and also gives some idea of how easy it is to construct such programs using high-level languages (in this case, Radio Shack BASIC). Why is it so easy to solve the problem this way? Because the computer can be regarded as a "third intelligence" which can hold data given to it by each player, compare it, and give out just that information which is needed at a particular stage in the game. The computer is a bit like a Gamesmaster in a fantasy role-playing game — I will come back to such games later on.

Readers will realise that this concept of a "third intelligence" has much wider application than naval search games, which are a small sub-category of wargames in general. Consider combat resolution, common to every wargame. One of the limitations that we take for granted is that, even if the strength and composition of the opponent's units has been concealed up to the moment of attack, they will have to be revealed so that the attack can be computed. In reality, of course, there may often be situations in which the original strength of the enemy remains unclear after the attack has been broken off, and to have exact knowledge of the losses he suffered is very rare. Similarly, one "books" an artillery barrage which is to be followed by an attack, not knowing how effective the barrage will be — perhaps it will fall on empty trenches, the main line of resistance having been pulled back a mile or so in the night. One does not call in the barrage gun by gun until the enemy reports that his strength has been reduced to 6, and that the attack may as well proceed, since we are using an odds-ratio CRT and cannot get better odds than 2-1.

Wargamers have put up with this sort of thing for many years, but there is no need for it. The computer can easily remember the original strengths of 50 initial units, and decrement these strengths whenever losses result from combat. When told to resolve a fresh combat between specified units, it will use the current strengths to calculate the correct ratio, roll a die (choose a random number) and announce the result — there's no need for either player to be told the residual strength of any of the units involved, and there are situations where one would want to withhold this information even from the player controlling the units in question (if he was directing the attack from a distance, and the designer wanted to simulate the difficulty a commander has in knowing what is going on at the front).

All wargamers tend to be amateur game-designers, and I can already imagine many readers turning to the games pile and bringing out some favourite but essentially unplayable game — like **"October War"** — and retrofitting it with a few short programs to supply the necessary features of limited intelligence to give the games a new lease of life. Is this wise?

It's tempting, but the effort involved may be much more than is apparent at first sight. The reason for this is that any wargame is a fragile network of interdependent game mechanics, and the original designer, knowing that there were certain aspects of the situation that he couldn't simulate effectively, factored them out of the mechanics. Perhaps he included their historical effects elsewhere, such as in the combat strengths of the units on one side, or perhaps he arranged the Victory Conditions so that the player who historically chose a certain strategy because of faulty intelligence, will tend to choose the same strategy in the game, and keep it on historical lines, in pursuit of Victory Points for fulfilling a particular set of conditions. Change one element of such a game, and the whole system may fall apart. Anyway, consider the game-scale that the designer chose for a particular battle, or even the choice of that battle rather than some other. Weren't these choices made because of the lack of suitable means of simulating limited-intelligence situations, so that only battles and campaigns in which these considerations were less than crucial have in the past been simulated at all? It seems more likely that the result of the impact of computer assistance on the wargame scene will be a crop of *new* games, rather than a series of patches on the old ones.

## Taking the dirt out of the mechanics.

Consider the outline program for double-blind search; how much of the players' time did it actually save? It depends on how many separate actions the players would have been required to perform in order to resolve the effects of a single search in a manual game. Perhaps the searching player has to roll dice to determine extent of cloud cover, make allowance for different search ranges, roll dice again to see if the searching planes were shot down, roll yet again to find out if they got their report back before they were shot down .... the computer can take care of all that, and very quickly, too. On the other hand it may be necessary for the players to spend time telling the computer things like where all the units are at the end of the movement phase, so that it can use this information at a later stage. In general, computer programs can absorb dirt from games with dirty mechanics — meaning lots of repetitive routines that each player must perform in every turn — but it is not at this stage clear how much time can be saved in this way, and, once again, if a designer wants to make full use of this facility he will need to take it into account from the earliest design stages, rather than tacking it on at the end.

**"Highway to the Reich"** is an obvious candidate for such treatment, and an analysis has been made of the playing time that could be saved by a set of game-assistance routines on the computer. The results are shown in the separate panel **"Highway to the Reich,"** containing details of the various phases in the Game-Turn and the time taken to complete them with and without computer assistance (p.10). **The reader will notice that for some phases, the players are estimated to need more time with computer assistance than without. This is of course because they have to spend time telling the computer things like which units are out of supply.** Nevertheless, if we assume that an average game will last about 80 turns, it looks as if computer assistance would allow us to complete the Campaign game in, say, a month, which may be tolerable; playing it manually, we would be stuck with the thing for about ten weeks!

This only scratches the surface, however. One of the chief irritations in "HWTR" is the cardboard mountains that build up around hotly-contested hexes. On analysis, the average large stack turns out to contain about 50% combat units and 50% strength chits. The computer will be happy to keep a list of all the combat units in the game (if it has enough memory) and, since it is already resolving combat and telling us the results in terms of strength losses, it may as well keep it's list up to date in terms of the current strength of each unit. These strengths would of course be used in future combats, and would be available to the friendly player on request; the information can be withheld from the enemy player if desired. The main point is that counter density on the map can be reduced by half, since we can now throw away 1200 uninteresting numbered strength chits. And, of course, the current strength of an attacking unit need no longer be entered into the computer during the Combat Phase. Generally, retrofitting of this kind is only going to be worthwhile if it is done on a wholesale basis, affecting every aspect of the mechanics of a particular game.

## Teaching the players the rules.

We have not yet isolated the most important point about computer programs of the kind described above. **"HWTR"** was designed as one of the first of a range of what **SPI** called **"HIMS"** games, by which they really meant games in which playability had been entirely sacrificed to considerations of "realism," which, in turn, really means detail. It turned out that there was a category of wargamers prepared to accept such games, and even try to play them. On the other hand it was clear that non-wargamers would never be enticed into the hobby by such monsters; if the game companies did not want their market to shrink steadily away, they had to go on producing simple games to attract newcomers. It seems that 3½ pages of rules is about as simple as a wargame can get .... and still there are many people who refuse to consider learning a game unless the rules can be explained in less than 5 minutes!

This dilemma, which has been the central problem for wargame designers from the earliest days, is now no longer an issue. If the computer can resolve combat for you it follows that you don't need to read that section of the rules concerning combat before starting play. The same goes for any other section of the game-mechanics that can be shoved into computer memory. For that matter, if the designer suddenly decides that he wants to make the combat mechanics twice as complicated, with twice as many odds modifiers and twice as many special rules that apply only to attacking Belgian engineers in Turns 2, 4 and 8, he can go ahead and do whatever he wants; the perceived complexity of the game, so far as the players are concerned, will remain unchanged, just so long as the extra modifications are "computable," that is, capable of being translated into algorithms the computer understands.

Of course it will be argued that the player *ought* to concern himself with the way that combat is resolved, in order to play skilfully. What we're trying to do, however, is give him enough information so that he can just get started on the game. Once he is playing it, he'll soon realise what he needs to know to play well .... and then he will want the computer to tell him what rules apply to the particular situation in which he finds himself.

The ability of the computer to do this means that it can be used to teach the players the rules as they go along, removing the necessity for them to spend a week poring over a 32-page rulebook before getting to Turn 1, and making the entire hobby much more attractive to outsiders. All the research carried out by **SPI** and other companies suggests that the complexity of wargame rules is the main reason why people shy away from becoming involved with them, or more precisely, they do not not want to spend several hours learning all the rules as a first step. In a game like **"Ney vs. Wellington,"** however, it's really only necessary to be familiar with a small section of the rules in order to carry out a particular Phase in the Game-Turn, and so the player wants to be referred to those rules and to no others. The computer can work out which rules apply to a particular contingency during play, and can also spot rules violations, keeping the player on the right lines during the learning process. Best of all, the computer would hold all the rules in memory together with examples of play and explanations of obscure points, and would quote the appropriate extract from the rules text at the right time. Unfortunately, it's not likely that the average domestic microcomputer will have enough memory capacity to hold so much text, since it's very expensive (in terms of memory consumption) to store normal text (known as "strings") in comparison with other forms of program line. Nevertheless, computers are good at searching and sorting operations, and the BASIC high-level language built into most micros is particularly good at carrying out such operations on string material as opposed to numeric data (much better, by the way, than some so-called "advanced" languages like Fortran, APL or even Pascal). So the possibility arises that wargames can be "sold" to a large section of the population who would not otherwise have been attracted to them. Dyed-in-the-wool wargamers may not care very much whether this happens or not, but this would be a very short-sighted attitude. One of the main problems facing the game companies at the moment is that games on unpopular subjects can never sell in large volume, and the temptation is to produce nothing but games about tank battles on the Eastern Front, since any such game, however appalling in quality, will always sell enough copies to break even, whereas a game on the Battle of Marathon would have to be of exceptional quality to stand a chance of making a profit for all those involved in its production. If the wargame market were to increase in size by a factor of 10, however, an entirely new set of principles would come into effect to govern the situation, since any game on any subject could in principle sell enough copies to be profitable. The question would then be whether a particular game was better than the competing titles in its subject-area, and we would begin to see game companies paying more attention to the actual merits of each design, rather than to a head-count of the number of people who expressed a desire in the Feedback to buy any game on a given subject.

Once the player is familiar with the mechanics of the game, he does of course wish to study the precise formulas used by the designer in quantifying the movement and combat abilities of the forces involved, so that he can optimise his play. He will therefore still need to be provided with all the charts and tables in which these details are laid out. The important point is that this secondary process, in which the game is studied in detail, can be separated from the primary learning process, which has to be undergone before the data contained in the game can be interpreted properly. At the moment, the wargamer is forced to combine these two activities, and the mental effort required to do this is the real reason that it is so difficult to attract people into wargames.

## Real-time games

A real-time game is one in which action is continuous, rather than being split up into Phases and Game-Turns. It's still possible to "stop the action" at any point, then resume play after studying the situation, but it isn't possible for one player to move his units while the other player looks on helplessly, nor is it possible for a player to carry out complicated manoeuvres with a small group of his units while all the others remain in a kind of stasis. All units that have been set in motion will continue to move in accordance with orders until told to do something else; all interactions between opposing units trying to occupy the same ground will be resolved whenever they occur, rather than being postponed to a separate combat Phase. From this description, it can be seen that the various attempts to SiMov game mechanics that designers have made in the past have all been intended to achieve the effects of real-time games within the framework of conventional wargame mechanics. Most players who have tried such games will agree that they achieve only a limited measure of success, not so much because of any intrinsic impossibility in creating a SiMov system that takes into account all the necessary features, but rather because the burden imposed on the players by a full-blown SiMov system is intolerable. The sort of people interested in tactical armour conflicts between 1941 and 1990 don't want to write out the movement path of every vehicle through every hex in every Turn.

Real-time games therefore belong in a subset of types of dirty mechanics; a computer-assisted wargame would give you SiMov without dirt, but writing the computer program would not be entirely straightforward. The map itself would have to be in the computer, in the sense that the computer was in a position to remind the player that a particular hexside was uncrossable by the type of unit that wanted to cross it; the computer must know all the characteristics of the map if it is going to record all the proposed moves of the units belonging to each player and then legislate on what happens as each unit attempts to enter each hex. It should be possible to get the map into the computer in this way, but no-one has tried it yet.

All these proposals seem to take more and more control of the game situation away from the players, giving it to the computer, which acts as a "third intelligence." Another way of looking at it, of course, is that the machine is the umpire between two opposing players or teams, and it ought to be born in mind that this is the original concept of the "War Game", as played out by military forces of all nations, on all levels from minor skirmishes traced on maps to the recent NATO exercises in West Germany. In all such games, a team of umpires has to mediate between the opposing forces in order to decide the results of each conflict, and can choose to release information selectively to each player concerning the overall situation. A microcomputer is ideally suited to fill this role, and we can anticipate that future computer wargames will be much more realistic (and objective) than any comparable exercises carried out by the Institute for Strategic Studies.

## Artificial Intelligence

In the final part of this article we consider more advanced applications of computers, including the idea that a computer can take one side in a two-player wargame, holding its own against a single human opponent. An attempt will also be made to survey the current situation in the computer wargame market and predict future trends, but progress is so rapid in this field that such comments are likely to become out of date very rapidly.

One statistic that has remained constant over the last six years or so is that more than half the wargames that are played are played solitaire. Some years ago it could be argued that this was mainly because players had difficulty in finding opponents, but, with the growth of various contact groups, this explanation has become less plausible. It seems that many people simply prefer to explore a combat situation by working out optimum moves for both sides, or else that they have a theory about optimum play for one side and would like to see if it works by playing it out against a conventional defense by the other side. A computer program which took one side in a wargame and allowed the human player to try to defeat it would attract a great deal of interest. A number of chess-playing programs for microcomputers of this sort are on the market, and in theory the task of creating a computerised Soviet defender in **"Panzergruppe Guderian"** should not be too difficult. In reality there are a lot of problems to be overcome, and the available literature on game-playing Artificial Intelligence programs is not much help. This is because the history of this discipline has for some years been dominated by the search for a chess program that can beat a Grandmaster and games of other types have tended to be ignored. Unfortunately a machine that plays chess teaches us nothing about either human or machine intelligence – all it does is remind us that the ability to play chess expertly is a specialised skill with little relevance to anything else.

One or two games have nevertheless come on to the market which allow a human player to oppose a micro in a combat situation. In **Fire & Movement** issues 17 and 18, Dave Minch discussed such games and reviewed **"Tanktics"**, a game in which the human player attempts to take a defended position from stationary units controlled by the computer. Recently, advertisements for **"Computer Bismarck"** have appeared in this country; in this game the player has the option of commanding the British air and naval forces against the Bismarck, which attempts to escape to the North Atlantic under computer control.

The first thing you discover when you try to write a computer program to simulate a human wargame player is that the two essential elements of a wargame, movement and combat, the first is much more difficult to program than the second. The procedures used in traditional wargames for resolving combat are already computer-compatible in the sense that the logic used by the players to execute the combat procedure corresponds to the logic of a high-level computer language. Arranging for the computer to move a hundred combat units into the optimum configuration is, however, quite a different matter, and defies analysis at present. Both the games mentioned above can therefore be regarded as special cases; in one, the machine's units don't move, and in the other, there is only one unit controlled by the machine, and it's movement is likely to have a random element anyway.

The most important difference between the two programs is the physical components. **"Tanktics"** uses conventional map and counters, whereas **"Bismarck"** attempts to dispense with them, putting the map up on the screen and telling the players to keep track of their units with log pads. In order to make clear the significance of this, it will be necessary to explain the main problem in computer software nowadays – piracy.

A computer program is normally sold on a cassette of exactly the same type as an audio cassette and indeed, the program can be "heard" as a series of high-speed data blips. Once it has been loaded into the computer from a cassette deck, a blank tape can be placed in the deck and the machine told to make a copy of the program to tape. If the company marketing the program has inserted machine-language routines in an attempt to prevent such copying, a deck-to-deck copy can always be made using a second deck. In general, then, there is no way in which piracy of programs can be prevented. Even if it could be established that copyright law applied to computer programs, most piracy involves giving a friend a free copy of a purchased program – very difficult to track down. Making a copy of a three-colour wargame map and die-cut laminated-board backprinted counter sheet is, on the other hand, hopelessly difficult unless you own a printing firm. All the physical components of **"Computer Bismarck"** apart from the tape appear to be easily photocopiable, whereas you would have to buy **"Tanktics"** if you wanted your own copy.

Piracy is such a threat to software houses that it seems possible that a proper industry in this field will never really develop. However, a genuine computer wargame, with map, counters and accompanying program, would not only be immune to piracy but would offer users a game of

| Budget | |
|---|---|
| up to £150 | **Sinclair ZX80. 1K user memory,** expandable. Fully programmable; some inconvenient aspects in use. Needs domestic TV, and cassette deck to save programs. Simple BASIC resident in board. |
| | Acord Atom. 2K user memory, expandable. Needs TV and cassette deck. 8K BASIC resident. |
| £300 | TRS-80 keyboard. 4K user memory, Level 1 (4K) BASIC resident. Expandable. Needs domestic TV and cassette. |
| £400 | TRS-80 keyboard. 16K user memory, Level 2 (12K) BASIC. Expandable, at some expense. Needs TV and cassette. |
| | Video Genie. Identical to above except that cassette is built in. |
| £600 | TRS-80, 16K Level 2, includes VDU and cassette deck as separates.<br>PET 2001. 16K memory, BASIC similar to TRS-80. VDU integral with keyboard, cassette deck as separate. |
| | Sharp MZ-80K. 6K memory with BASIC (14K) loaded (with no interpreter, 20K). Integral cassette and VDU. |
| £900+ | For the extra money you can have (a) colour display, as on the Apple; (b) more memory; or (c) a disc drive, replacing the cassette and operating many times faster. |

Note: the excellent products of Nascom have been omitted as this firm is currently in receivership. Exidy no longer market the Sorceror but it is available. Many new models are appearing at the bottom end of the market.

much higher quality than the average, arcade-related "computer game" which is already beginning to give micros a bad name. In sponsoring **"Computer Bismarck"** in its present form, **Avalon Hill** may have made a bad marketing error.

## Limited Intelligence

Suppose that we have written a program to allow the machine to operate the Soviet forces in **'Panzergruppe Guderian'**, as mentioned earlier; the computer will be following one of a number of possible strategies, such as hasty withdrawal, forward defense, or withdrawal followed by a counterattack in the South. The human player will also be following a coherent strategy. The human player will have to inform the computer of all his moves, of course, and the machine will have to list the unit movements that it wants to make on the map each Turn. Each side will therefore be aware of the exact disposition of the other's units, as in a manual wargame. Can this problem of perfect intelligence be tackled? There should in fact be no great difficulty, since the computer can keep an accurate record of the human player's units in order to give assistance with the play of the game, pretend that it doesn't know where they are when calculating its own moves, and give the player partial information about its own units, thus apparently exhibiting a split personality. If on the other hand the player wants to explore one particular strategy open to the Soviet player in **"PGG"**, he can specify that the computer will adopt that strategy and then examine alternative reactions to it.

## Real-time Games

It's a bit of a paradox that you have to spend perhaps ten minutes to work out what happens in a few seconds of combat in **"Air War"**. Generally, it's a pity that as the level of detail in a game increases, so does the amount of data that the players have to feed into the game system. One could imagine a micro piloting three Mig-25's in real-time and asking for the moves of the human player's aircraft, but there is really no way in which human responses could keep up with the computer's speed. This is partly because of the primitive nature of peripherals – if the input could be spoken voice, the human player might have a chance. A more promising approach is to place the human player in the position of a Corps or Division commander, feed him with minute-by-minute updates of the enemy's activity, and ask him to respond with a continual stream of orders to his various units. This, in fact, was the original idea behind **SPI's "NATO Division Commander"**, now being produced as an entirely manual game, and when wargames which are genuine applications of Artificial Intelligence techniques finally appear, this sort of game will probably become quite common.

## PBPM

Play-By-Mail normally involves writing out your moves, putting them in an envelope and waiting perhaps four days for them to reach your postal opponent and then come back to you with his moves. In the past this sort of thing has appealed only to people with a lot of patience, but there is now the possibility of Play By Phone Modem, a much more satisfactory activity. A Phone Modem is a gadget which connects your micro to a normal Post Office telephone handset, allowing the transmission of data through the public telephone system direct to any receiving micro equipped with a similar device. It is actually a lot cheaper than letter mail, since the machine can be programmed to send the data in the middle of the night if necessary, to take advantage of cheap rates, and material is transmitted relatively quickly, typically taking only a few seconds. The capital investment in a modem could therefore be recovered eventually and quite large games could be played through in a reasonable period of time.

With the recent relaxation of the Post Office's monopoly in this area, such devices will spread rapidly, and we are likely to see the appearance of several competing organisations offering nationwide data networks; the user patches his micro into the system and can then communicate with any other users in the network, allowing multi-player games. Two of these networks are already operating in the States, but be warned; one of them, The Source, has run into financial problems and the moral seems to be that one should wait a

## DOUBLE-BLIND SEQUENCE

The program shown below (in Tandy BASIC) will accept input of hexes to be searched by each player, then display on the screen the hexes in which enemy units have been located.

```
10 DEFINT S, J, U, T, X, Y
20 DIM JP(15), US(15), SU(15),
SJ(15): TU=1
30 'PROGRAM FOR MOVEMENT PHASE
HERE
120 JL=704: UL=960
130 FOR X=1 TO 10: SU(X)=0:
SJ(X)=0:JP(X)=0: US(X)=0:NEXT X
140 CLS: PRINT@0,"TURN";TU;:
 PRINT@64,CHR$(15);
200 FOR X=1 TO 15
210 INPUT"HEXES SEARCHED BY U.S.
 PLAYER (9 IF NO MORE)";X$
212 SU(X) = VAL(X$)
215 IF SU(X)=9 THEN 230
220 NEXT X
230 FOR X=1 TO 15
240 INPUT "HEXES SEARCHED BY
JAPANESE PLAYER (9 IF NO MORE)";
X$
242 SJ(X) = VAL(X$)
245 IF SJ(X)=9 THEN 300
250 NEXT X
300 CLS: PRINT@640, "JAPANESE
 UNITS LOCATED IN HEXES:-";
310 PRINT@896, "U.S. UNITS
 LOCATED IN HEXES:-";
400 FOR X=1 TO 15
410   FOR Y=1 TO 15
420   IF SU(X)=9 THEN 450
425   IF JP(Y)=0 THEN 440
430   IF SU(X)=JP(Y) THEN GOSUB
       1000
440   NEXT Y
450 NEXT X
500 FOR X=1 TO 15
510   FOR Y=1 TO 15
520   IF SJ(X)=9 THEN 550
525   IF US(Y)=0 THEN 540
530   IF SJ(X)=US(Y) THEN GOSUB
       2000
540   NEXT Y
550 NEXT X
580 A$=INKEY$:IF A$="" THEN 580
600 TU=TU+1: GOTO 120
1000 PRINT@JL,JP(Y);
1010 JL=JL+6
1020 RETURN
2000 PRINT@UL,US(Y);
2010 UL=UL+6
2020 RETURN
```

This is intended as part of a larger program in which the players tell the computer the hexes occupied by their units, earlier in the Turn, then proceed to the search phase. If there is only one keyboard and screen, it will be necessary for one player to look away while the other is entering data, although it can be arranged that the information being entered doesn't appear on the screen, which helps.

Line 20 reserves space for four tables in computer memory. These are called vectors and in this case each has only one "row," with fifteen spaces to hold information – in this case, hex numbers. Vector JP holds the location of Japanese units, US holds the location of U.S. units, SU will hold the numbers of hexes being searched by the U.S. player, and SJ will hold hexes being searched by the Japanese player. Line 120 simply defines the area of the screen at which messages will appear, making for a tidy display. Line 130 zeroes the values held in the vector locations, left over from the last Turn. SU(3) is the name of the third location in vector SU, and so on.

In line 140 we Clear the Screen and print the Turn number. Line 200 sets up a loop which will repeat itself 15 times. Each time, the U.S. player is asked to enter the number of a hex he is searching, or type "9" if he has come to the end of his list. The INPUT statement will cause the computer to display the quoted text and then hang around until the user types something and presses the ENTER key. Line 212 is a way of coping with the possibility that the user types a letter by mistake, but more programming would have to be added to analyse the mistake, display the appropriate error message, and cancel the mistaken input.

Each time round the loop, X is incremented by one and so the input is stored in the next location in the SU vector. Line 215 allows escape from the loop, and causes execution to jump to line 230. Line 230 starts the same input procedure for the Japanese player, so there should be a CLS statement here to wipe out the screen display showing the U.S. player's search data. The routine from 300 to 550 then displays all hexes which were searched and contain enemy units, but no others. If we wanted, we could split this into separate displays for the two players. Lines 400 to 450 are a nested loop in which the computer takes the value in location 1 of SU and compares it with the value in each location of JP in turn, then does the same thing for SU(2), and so on. Each time it finds a match it branches to the subroutine at line 1000, printing the value on the screen and then returning to the end of line 430. The process is repeated for the SJ and US vectors, using the 2000 subroutine.

In line 580 the computer will hang fire until any key is pressed, then it will drop through to line 600, increment the Turn number, and go back to repeat the cycle, which would normally mean starting with the movement phase of the next Turn.

This program occupies around 1K, but only because it uses short program lines for clarity. The logic could also be considerably condensed and then it would fit easily into a ZX80 (re-written in ZX80 BASIC), with room for additional routines.

A number of additional routines should really be added, preventing the program from crashing if the wrong type of data is entered, preventing one player from obtaining data entered by the other, and checking the data to make sure it conforms with the rules. None of this presents any real problems. It would be neater if we could have two VDU's back to back, displaying material on one screen or the other; whether this is possible depends on the internal architecture of the computer being used.

## FOLLOW-UP

If you have any comments on the issues discussed in this article, or would like to know where to find more detailed information on hardware or software, you can write to:

**Mike Costello**
**17 Langbank Avenue**
**Rise Park**
**NOTTINGHAM NG5 5BU**

I'll be happy to give any advice that I can.

## HIGHWAY TO THE REICH

Estimated time taken to perform each phase in an average Game-Turn.

| Phase | Manual | Computer-assisted |
|---|---|---|
| 1. Allied Command | 10 | 8 |
| 2. German Organization | 4 | 5 |
| 3. Allied Conditional Fire | 30 | 9 |
| 4. Allied Conditional Move | 10 | 9 |
| 5. German Conditional Fire | 10 | 3 |
| 6. German Conditional Move | 7 | 6 |
| 7. Allied Free Fire | 45 | 14 |
| 8. German Command | 4 | 4 |
| 9. Allied Organization | 9 | 10 |
| 10. German Conditional Fire | 10 | 3 |
| 11 German Conditional Move | 7 | 6 |
| 12. Allied Conditional Fire | 30 | 9 |
| 13. Allied Conditional Move | 10 | 8 |
| 14. German Free Fire | 25 | 8 |
| Totals | 3 hours 31 mins. | 1 hour 42 mins. |

little while before committing oneself to one system or another. In the end, though, the advantages of such a set-up are so enormous that it is bound to supersede postal play altogether. It is really a natural development of the history of wargaming; first you couldn't find an opponent because there weren't any other wargamers around, then when there were more gamers there were more games and no-one was interested in the games you were interested in. Now at last you can pick out the opponents who share your interests, however far away they are, all from the comfort of your own home.

## Fantasy

Fantasy gaming is the fastest-growing section of the hobby though many confirmed wargamers find this inexplicable. Most of these games seem to share the same common factors: three or more players, an umpire or Gamesmaster and a game-structure of progressive discovery on the part of the players as they explore their environment – rather like the **"Adventure"** computer games, which are said to derive from **"Dungeons & Dragons,"** although it may well be that the influence was the other way round. The key demand that players of these games have is that they be able to enact roles corresponding to the characters in the game and also that the games be open-ended in the sense that surprises can happen however often the games are played and re-played. Obviously this is difficult to arrange, and game-systems like **"Dungeons & Dragons"** are now surrounded by an elaborate network of reference books with lists of new monsters and spells, fanzines with new scenarios and earnest discussions of the legitimacy of additional features to be added to the basic system. The unwillingness of players to be tied down to complete, legalistic rules seems related to the desire to incorporate the unexpected into the game, and the function of the Gamesmaster is not so much that of an umpire but rather that of a human Random Events Table, dishing out local colour to each player in turn.

This argument, tied to the apparent difficulty of finding good Gamesmasters (it's a boring job), leads to the conclusion that the ideal fantasy role-playing game would be refereed by a computer which would inject the necessary element of the unexpected and also take care of the tedious aspects of the mechanics, like repeated rounds of single combat. The program would have to be sensitive to the narrative flow of a particular game, which is to some extent under the control of the players, so it could not be just a passive game-assistance routine. A computer can only output a version of the same material that was input to it, of course, so we are talking about successive permutations of the same data, but this is also a feature of many F&SF games at present. But it is not at all clear how a group of players would react to the intrusion of a computer into their fantasy, and perhaps they would not find it consistent with the kind of game they are trying to construct. As in many other cases, we shall have to wait until someone tries it.

## Summary

At the end of 1978 there were perhaps 5000 microcomputers in the U.K.; one year later the figure had risen to around 20,000 and the rate of increase is continuing to accelerate. The increase in the quantity and quality of software, however, has not matched this growth, and there are many potential applications for leisure use of micros in the domestic market. In this article it has been suggested that computer wargaming will be a large (and lucrative) section of this leisure market just as soon as the designers get their priorities sorted out and establish what is possible and what isn't. At the time of writing, **Avalon Hill** have launched five computer wargames, **SPI** will re-introduce **SpiBus**, and a number of smaller companies have declared a stake in this prospective market. The buyer should be warned: just as there are some bad wargames around, so there is some really terrible software, and a combination of the two would be a pretty grim prospect. On the other hand, a game which had been designed from scratch as a genuine synthesis of conventional wargames and computer assistance should represent a new lease of life for the hobby.

# MICROCOMPUTERS IN WARGAMING

**Stop Press**

Quite a few changes have come about since this article was first written, and this section is intended to bring readers up to date (as at November 1980).

A buyer was finally found for Nascom, and the company will continue trading, although it is not clear whether the full range of products will continue to be supported. 5000 ZX80's are now being sold each month and it now seems that this machine will double the micro market by the end of 1980. Expansion sets are available for purchasers who are running out of RAM.

Tandy have slashed the price of the 16K Level 2 TRS-80 to £469 as a special offer. It may become the permanent price and represents incredibly good value which may force a further round of price cuts among its competitors. Tandy are now selling equipment for the connection of micros through the 'phone lines and British Telecom are developing a "black box" to allow equipment from different manufacturers to be linked together.

The Sharp pocket computer has made its appearance at less than £100. This is a genuine computer although it has only about 2K of memory and a restricted screen display. It doesn't require a TV but a cassette deck for saving programs is about £25 extra. Tandy are selling it under their own logo for an inflated price.

Four of Avalon Hill's computer games are being advertised in the US on a mail-order basis since wargame stores are not very interested in them. They are, in fact, conventional arcade-type games except that they lack graphics and require the solitaire player to do a great deal of routine work. **"Computer Ambush"** is also available and looks like a simple form of **"Squad Leader"**, but appearances may be deceptive. Automated Simulations is a new company aiming for the fantasy market with a linked series of real-time solitaire games derived from **D&D**. With all games of this structure, of course, the trouble is that once the player has 'cracked' the game-system he has solved the puzzle and has no further interest in the game.

SPI rightly point out that these products are eating into each other's markets rather than generating a new one. Time will tell whether a more impressive range of software will become available.

# STARFIRE

**PAUL KING**

Minigames, whether we like them or not, are proliferating at a fantastic rate. Admittedly some subjects lend themselves to the small mapsheet/short scenario format – perhaps the most obvious being the field of Science Fiction. Tactical S.F. games abound covering ship to ship combat, often in great depth. Few games, however, allow large space fleet actions using individual ships while keeping game mechanics to a minimum. **Star Fleet Battles** by **Task Force Games** is one example of a minigame with much detail in it's workings. By the same token, another game in the **Task Force** series allows large actions, yet gives individual ship counters and sufficient detail for an extremely rapid game.

**Starfire** is it's name, and the game consists of a rule booklet, including combat tables, a hex grid, and a counter sheet with over a hundred pieces to make up three separate alien fleets. The rules cover slightly less than half the booklet and are extremely simple to learn. Each game turn consists of movement and fire; however, prior to this is an Initiative Determination Segment i.e. whoever throws the highest die number has the initiative for that game turn. Movement is on a hex by hex basis, one player moves his counters one hex, then his opponent moves his units one hex and so on up to the limits of movement. Since no ship ever moves more than six hexes per turn, this is not a particularly arduous task. The player with the initiative always moves last, thus knowing roughly what his opponent will do. Manoeuvring ships involves using a turn mode which means moving so many hexes, usually two or three, before making a 60° turn, as in **Star Fleet Battles.**

Combat, like movement, is sequential with one player firing the weapons of one ship, then his opponent fires and so on; in this segment the player with the initiative always fires first. Hitting a ship is dependent upon the weapon type and range. If the sum of two dice is less than or equal to a number on the hit chart, the ship has been hit. As for damage, well, some weapons only cause one point of damage, e.g. guns, whilst others have an increasing effect as range decreases, e.g. laser beams.

The system of movement and firing can be seen to be quick and simple and ship information recording is another simple matter. A typical escort ship will be written down as SAIGIRII – which tells me this ship has one shield, one armour unit, one gun, one missile launcher and four engines, and will thus move four hexes per turn. Every time a hit is sustained, one letter is crossed out, moving from left to right. So during a battle this lowly escort might limp off the board as ~~SAIGI~~RII – moving two hexes per turn and perhaps firing a missile at any target which presents itself. As ships increase in size, so does the string of letters, and the range of vessels available is quite comprehensive with cruisers, battle cruisers, destroyers, frigates and even super dreadnoughts and starbases!

The ten scenarios available build up an S.F. rationale from first contact to a massive fleet action by scenario eight. Scenarios nine and ten introduce a third player who is antagonistic to both the earlier participants, who in turn still dislike each other. These last two battles are no "Waterloo" with a friend coming eventually to reinforce a player, but are subtle blends of distrust, aggression and fear as foe attacks foe. Not only do the scenarios steadily increase the number and size of vessels, they are a well constructed method of gradually introducing the various weapons and defences. The overall "feel" is that of the Lensman or Skylark S.F. novel range by E.E. "Doc" Smith, with new and fearsome weaponry at each encounter. Initially weapons are confined to guns, missiles and lasers. Next we are given hybrid gun/missile launchers, then Point Defences to stop incoming missiles, then tractors, shear planes, force beams, primary beams and the list goes on. From playing various scenarios it becomes apparent that they are often biased in favour of one player who will have the new wonder weaponry, but this does not make the playing any less enjoyable. In fact a slight modification in ship type often helps, as does the addition of an extra ship to the ailing side. Two scenarios are directly linked, one involves the breakthrough of an enemy ship, the other the damage it inflicts on shipping lanes. Unfortunately set-up rules are missing for the latter (this being the only defect of the entire game I could find) but can easily be improvised. Talking of modifications naturally brings us to the optional rules. Most games these days seem to provide optionals, at times I often wonder why, for many seem pointless or increase bookeeping to a nuisance level and can detract from the game. **Starfire** is a clean, simple game, and the optional rules maintain this, adding to the game in a few ways but not detracting from it. One such rule is Datalink in which, for example, the missile fire from a few ships can be released as a volley at a specific target, overwhelming it's point defences. Of course this requires ships to act in squadrons, thus altering and refining tactics. Another useful rule concerns the mapsheet which is identical to that of **Star Fleet Battles,** and can be cut to provide a geomorphic

Continued on page 23