GLADIATORS

Simon Goodwin

Suitable for use in its own right or as a routine within a major Adventure game, this program should prove easily adaptable

In the last year there has been an upsurge of interest in microcomputer games of the 'Adventure' variety. In the field of detailing the environment in which adventures take place these computer games have been quite successful, but where they tend to fall down is in their system for simulating battles, a crucial part of many a dungeon campaign! This article describes a BASIC program that could be modified to simulate hand-to-hand combat in such games, and can also be fun in its own right.

Personality Profile

'Gladiators' is a game suitable for most microcomputer systems. The player has control over one gladiator, and the computer controls the other, in a fight to the death. Each combat is rated for strength, constitution and dexterity. Strength represents the ability of a gladiator to do substantial damage to his opponent (eg kill him) — the greater his strength, the more telling each blow will be. Constitution reflects general fitness and staying power; the amount of punishment a fighter can take before shuffling off this mortal coil to join his tribal ancestors. Dexterity is an expression of co-ordination — the greater a gladiator's dexterity, the more likely it is that he will be able to dodge a blow.

Tactics Made Simple

In essence, 'Gladiators' is a computer-moderated guessing game in which the player aims to beat the machine by making a succession of tactical decisions. The human habit of assuming that anything that can make decisions must be intelligent, adds to the interest of the game, since the semi-random manoeuvres of the computer's gladiator can often seem to be more planned than they really are. On the basis of information about the capabilities of his opponent and himself, and the physical distance between the fighters, he decides upon one of six manoeuvres:

BASH — a blow to the enemy's head. The smaller the distance between the combatants, and the greater the attacker's strength, the more damage this will do to an opponent. If a blow is not dodged it will result in blows equal to the attacker's strength at a range of one, half that at range two, a third at range three, and so on. The number of blows is always rounded up to the nearest whole number, and brings about an equal decrease in the enemy's constitution. If constitution falls below one a fighter is dead.

DUCK — an attempted dodge of a blow to the head. When the dodge is attempted a random decimal number between 0 and 20 is worked out. If the result is less than the defender's dexterity rating the duck is successful, and the opponent's Bash has no effect. Of course if you weren't bashed in the first place you'd be wasting your time! One popular mistake is a Duck in response to a Stab in the guts. Not a recommended strategy!

CLOSE — a move towards the opponent. This brings about a decrease in the 'range' between the two fighters unless the other gladiator Closes or Retreats (see below). Range can never be less than one, and decreases by one unit at each Close by either combatant.

RETREAT — a move away from the enemy (cf Bloodnok, Major, Coward and Bar). Range increases by one for each retreat, hence if the player Retreats and the computer Closes in a turn, range will be unaffected. Maximum value is four, hence it is impossible to get completely out of range of your opponent. This rule may seem unfair, but it reflects the problem of players opting to avoid battle — in the Circus Maximus the Emperor might be expected to take a dim view of such a stratagem.

STAB — similar to a Bash, but a blow to the body of the opponent. It cannot be avoided by a Duck, and results in the same number of hits as a Bash if it isn't dodged.

PARRY — a dodge of a blow to the body, equivalent to a Duck for a Bash and used to stop the effects of a Stab. The higher your dexterity the more likely it is that the Parry will be successful.

Modus Operandi

The computer works out its moves without reference to yours (so they are effectively simultaneous decisions), using a combination random and rudimentary planned approach. The higher the total of the machine's 'attributes' (Constitution, Strength, Dexterity) the less likely that the computer will opt to Close or Retreat. If it does decide to carry out such a manoeuvre it will always Close if range is four, the maximum, and Retreat if range is one, otherwise making a 50:50 random choice. If the computer decides to stay its ground it will choose randomly which of the other four manoeuvres to carry out, making a decision influenced by its dexterity rating. This system is easily modified (lines 260-340 in the listing) to allow the computer to formulate a more 'intelligent' strategy — at the moment it is given an advantage of higher attributes than the player to make up for its moronic style of combat, but with practice and a little luck it is possible to defeat it half a dozen times in a row before you bite the dust. There is ample scope for tinkering in the Machine Move routine, but if it is improved the semi-random starting values of the attributes should be adjusted. Alter the 'recovery' value if you find that your character invariably loses his second battle — this value represents the effect of extra training and the healing of wounds. It might be interesting to give the player the choice of different types of training — the fixed number of recovery points could be optionally used to increase dexterity or strength instead of constitution. Upper limits of 10 for strength and 19 for dexterity would probably be a good idea if this system was used, otherwise a lucky gladiator could end up more like a Roman God, capable of dodging every

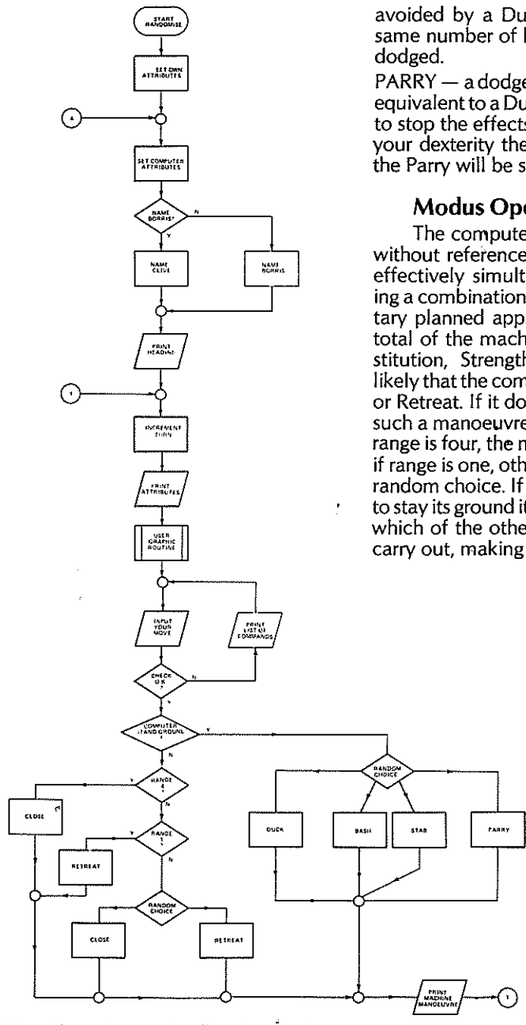

Fig.1. The main operating flowchart for the game.

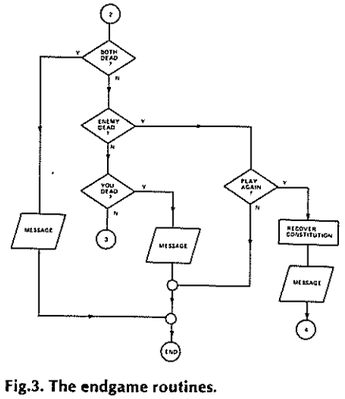

Fig.3. The endgame routines.

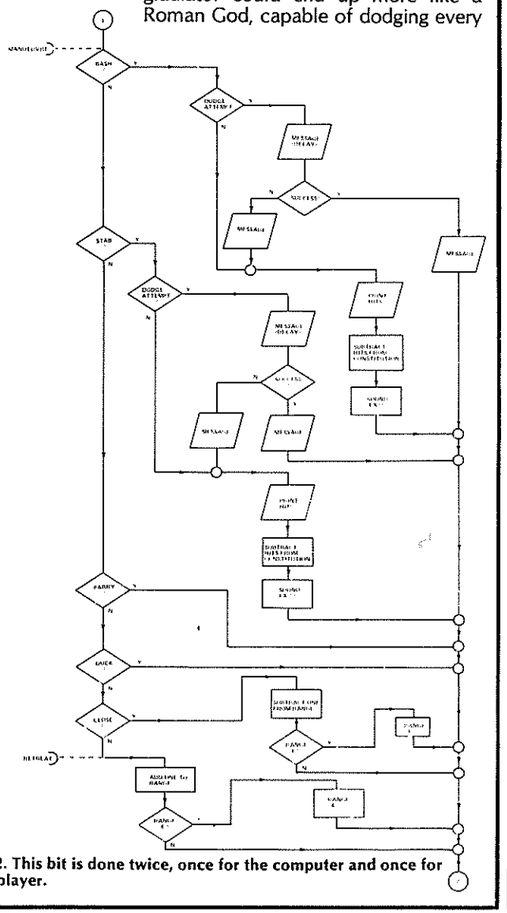

Fig.2. This bit is done twice, once for the computer and once for the player.

attack and killing off the opposition at a blow from long range.

The player tells the machine what move it has selected through his reply to the MANOEUVRE? command. If the first letter of the response is B(ash), S(tab), P(arry), D(uck), R(etreat) or C(lose) the appropriate tactic is carried out. If any other character is used at the start of the response a scolding message appears followed by a list of the valid instructions, followed by the MANOEUVRE? question once again. An entry of ? has the same effect, but without the message.

Modo Et Forma

The BASIC used in the program is slightly non-standard but should be easy to convert for most microcomputer implementations of the language. All the variables except M are integer types, marked by the % symbol in the listing. If a decimal is assigned to one of these it is rounded down to the nearest whole number. Omit the % symbols if your BASIC does not allow integer variable types, but bear in mind that you will need to put in some INT( ) statements to use the program with floating point numbers. The function Y = RND returns a random decimal between zero and one, and the random number sequence is 'seeded' to be different at each playing by use of the RANDOMIZE statement at the start of the program. The SLEEP 2 instruction (for example) generates a time delay of two seconds in the running of the program, and can be replaced by a FOR...NEXT loop or omitted altogether by the impatient.

To make the program as portable as possible few graphics or print formatting routines have been included in the version of the game listed. The text graphics used should work without modification on any 80 character wide display with a TAB command referenced to the left-hand screen edge. Semicolons in PRINT statements indicate that elements separated by the symbol should be printed without intervening spaces — commas space items out to the start of successive 'TAB fields', each of which is eight character-positions wide. The STRING$ (X,Y) command returns a string of X characters of ASCII code Y — for example STRING$ (5,42) returns "*****".

What Goes Where

Variables used in the program listing are as follows:

S% Player Strength
C% Player Constitution
D% Player Dexterity
U% Your move No
R% Range between fighters
T% Computer Strength
N% Computer Constitution
A% Computer Dexterity
M Computer move No
P% Turn No
G%, L%, V% General-purpose
B$ Commands
N$ Enemy name
A$ Reply variable

The program flowchart has been split into three sections to ease understanding. The middle section is executed twice, once for the computer's move and once for the player's. The section of program uses a computed GOSUB to conform with the tactical instructions, but this has to be shown on the flowchart as a series of Yes/No branches. The listing alternates two names for the gladiators — Borris and Clive (dedicated to a certain engineer...). For extra variety a more extensive table of names could be incorporated.

Aggressive Adventures

The 'Gladiators' program could be easily built into a home-made BASIC adventure game, to add variety. Build up an array of names and attributes of the characters in your game, and call 'Gladiators' up as a subroutine when required. Use a string array to contain the commands if you want to avoid upsetting DATA pointers in a big program. Alternatively a room could be set up using the 'Circus Maximus' theme, with add-ons such as the Imperial thumbs-down and so forth.

May the best man win...

Program Listing

```
 89  REM**INITIALISE
 90  RANDOMISE
100  S%=5:C%=9+RND*3:D%=8+RND*6
105  N$="BORRIS"
110  T%=5+RND*6:N%=9+RND*5:A%=9+RND*4:U%=5:M=5
115  PRINT:PRINT:R%=4:IF LEN(N$)=6 THEN N$="CLIVE"
     ELSE N$="BORRIS"
120  PRINT STRING$(33,35);
130  PRINT " GLADIATORS ";
140  PRINT STRING$(33,35)
149  REM**LOOP START
150  PRINT:PRINT:P%=P%+1:PRINT"[2 SPC]TURN NUMBER
     ";P%
160  PRINT TAB(23);"STRENGTH[3 SPC]CONSTITUTION
     [3 SPC]DEXTERITY":PRINT
170  PRINT"YOU : ",,S%,C%,D%:PRINT
180  PRINT N$;" :",,T%,N%,A%:PRINT
190  RESTORE:GOSUB 3000:V%=0:U%=0
200  INPUT"MANOEUVRE ";A$
210  IF LEFT$(A$,1)<>"?" THEN 220
215  FOR G%=1 TO 6:READ B$:PRINT B$,:NEXT G%:
     PRINT:GOTO 200
220  RESTORE:FOR G%=1 TO 6:READ B$
230  IF LEFT$(B$,1)=LEFT$(A$,1) THEN U%=G%
240  NEXT G%:RESTORE
250  IF U%=0 THEN PRINT"VALID COMMANDS ARE",:GOTO
     215
259  REM**MACHINE MOVE
260  M=RND*(55+F%/2):F%=T%+N%+A%+15
270  IF M>F% THEN 320
280  M=RND:IF M*30>11+A% THEN M=1:GOTO 350
290  IF M>.5 THEN M=2:GOTO 350
300  IF M*36>A% THEN M=3:GOTO 350
310  M=4:GOTO 350
320  IF R%=1 THEN M=5:GOTO 350
330  IF R%=4 THEN M=6:GOTO 350
340  M=INT(RND*2)+5
350  PRINT:PRINT N$;" RESPONDS WITH A ";
360  FOR G%=1 TO 6:READ B$:IF G%=M THEN PRINT B$:
     PRINT
370  NEXT G%
379  REM**THE BATTLE
380  ON M GOSUB 800,850,900,950,1000,1050
390  ON U% GOSUB 500,550,600,650,700,750
399  REM**COMBAT RESULTS
400  IF C%>0 OR N%>0 THEN 410
402  PRINT"OH DEAR, YOU'RE BOTH MORTALLY
     WOUNDED."
405  PRINT"BETTER LUCK NEXT TIME.":GOTO 10000
410  IF N%>0 THEN 420
412  PRINT N$;" FALLS TO THE GROUND AND EXPIRES.";
415  PRINT" WELL DONE, WANT ANOTHER GO ?":GOTO 440
420  IF C%>0 THEN 150:REM**NEXT TURN
425  PRINT"SORRY, BUT IN A FIT OF ENTHUSIASM ";
430  PRINT"YOU SEEM TO HAVE SNUFFED IT."
435  PRINT"YOU SURVIVED FOR ";P%;" TURNS.":GOTO
     10000
440  INPUT A$:IF LEFT$(A$,1)<>"Y" THEN 10000
450  G%=RND*7+2:PRINT"YOUR CONSTITUTION RECOVERS
     BY ";G%
460  C%=C%+G%:L%=0
470  GOTO 110
499  REM**YOU BASH
500  IF M<>4 THEN 530
505  PRINT N$;" ATTEMPTS TO DODGE - ";
510  SLEEP 5
520  IF RND*20<A% THEN PRINT"AND SUCCEEDS.":GOTO
     545
525  PRINT"AND FAILS."
530  PRINT:PRINT"HITS ON HIM ARE ";
535  G%=(S%/R%)+.999:PRINT G%,:N%=N%-G%
540  GOSUB 2000
545  RETURN
549  REM**YOU STAB
550  IF M<>3 THEN 575
555  PRINT N$;" TRIES TO DODGE - ";
560  SLEEP 2
565  IF RND*20<A% THEN PRINT"AND SUCCEEDS.":GOTO
     590
570  PRINT"AND FAILS."
575  PRINT:PRINT"HITS ON HIM ARE ";
580  G%=(S%/R%)+.999:PRINT G%,:N%=N%-G%
585  GOSUB 2000
590  RETURN
599  REM**YOU PARRY
600  RETURN
649  REM**YOU DUCK
650  RETURN
699  REM**YOU RETREAT
700  R%=R%+1:IF R%>4 THEN R%=4
710  RETURN
749  REM**YOU CLOSE
750  R%=R%-1:IF R%<1 THEN R%=1
760  RETURN
799  REM**ENEMY BASH
800  IF U%<>4 THEN 830
805  PRINT"YOUR ATTEMPTED DODGE - ";
810  SLEEP 3
820  IF RND*20<D% THEN PRINT"SUCCESS!":GOTO 840
825  PRINT"NO LUCK."
830  PRINT:PRINT"HITS ON YOU ARE ";
835  G%=(T%/R%)+.999:PRINT G%,:C%=C%-G%
840  GOSUB 2000
845  RETURN
849  REM**ENEMY STAB
850  IF U%<>3 THEN 875
855  PRINT"YOUR ATTEMPTED DODGE - ";
860  SLEEP 4
865  IF RND*20<D% THEN PRINT"SUCCESSFUL.":GOTO 890
870  PRINT"WITHOUT SUCCESS."
875  PRINT:PRINT"HITS ON YOU ARE ";
880  G%=(T%/R%)+.999:PRINT G%,:C%=C%-G%
885  GOSUB 2000
890  RETURN
899  REM**ENEMY PARRY
900  RETURN
949  REM**ENEMY DUCK
950  RETURN
999  REM**ENEMY RETREAT
1000 R%=R%+1:IF R%>4 THEN R%=4
1010 RETURN
1049 REM**ENEMY CLOSE
1050 R%=R%-1:IF R%<1 THEN R%=1
1060 RETURN
1999 REM**SOUND FX?
2000 ON G% GOSUB 2020,2030,2040,2050,2060,
     2070,2080,2090
2010 PRINT:RETURN
2020 PRINT"COUGH...":RETURN
2030 PRINT"GASP....":RETURN
2040 PRINT"YAEGH...":RETURN
2050 PRINT"EUFFF...":RETURN
2060 PRINT"REUCH...":RETURN
2070 PRINT"TAERCK..":RETURN
2080 PRINT"ARGHHH..":RETURN
2090 PRINT"@@#!@<<!":RETURN
2999 REM**VISUAL DISPLAY
3000 PRINT:PRINT"RANGE ";R%;".";TAB(29-R%);
3010 IF U%=4 THEN PRINT"* ";ELSE PRINT" *";
3020 IF M=4 THEN PRINT TAB(30+R%);" *";ELSE PRINT
     TAB(30+R%);"* ";
3030 PRINT TAB(64);"<- SEPARATION."
3040 PRINT TAB(30-R%);"]";:IF U%=3 THEN PRINT"}";
3050 IF U%=2 THEN PRINT"\";
3055 IF U%=1 THEN PRINT"~";
3060 IF M=1 THEN PRINT TAB(29+R%);"~";
3065 IF M=2 THEN PRINT TAB(29+R%);"/";
3070 IF M=3 THEN PRINT TAB(29+R%);"(";
3075 IF M>3 THEN PRINT TAB(29+R%);" ";
3080 PRINT TAB(30+R%);"{":PRINT TAB(30-R%);"[";
     TAB(30+R%);"]":PRINT
3090 RETURN
8999 REM**LIST OF COMMANDS
9000 DATA BASH,STAB,PARRY,DUCK,RETREAT,CLOSE
9999 REM**THAT'S ALL FOLKS
10000 END
```