



Introduction

This guide has been written to help you to write marketable software. The market for home computer software has now become quite sophisticated, and even an accomplished machine-code programmer who has written a top-quality game may fall prey to the intransigent nature of today's software marketplace.

We hope that after reading this guide you will have gained some of our accumulated knowledge, and that this will enable you to produce a piece of software that is not only a high quality product but also brings you the financial rewards that your programming deserves.

By Christopher Payne: Marketing Consultant, Superior Software Ltd.

Copyright © Superior Software Ltd., 1986
All rights reserved.



Selecting the Target Computer

Due to the cluttered state of the software market with too many games competing for a teenager's pocket-money, a game certainly has to be distinctive to make any real impression. The Spectrum marketplace is quite overcrowded with many software houses preferring to concentrate on the Commodore 64 and Amstrad markets. Very few companies support the Atari (300,000 computers sold), but rather more release programs for the BBC Micro (500,000 computers sold) and the Electron (300,000 computers sold). In the Spectrum and Commodore markets the "product life-cycle" of games is rather short, sometimes as little as four weeks, but in the BBC and Electron markets certain programs can go on selling well for months.

Submitting Games



Assuming you have written a game, you will need to find a suitable software publisher. You could publish the game yourself; but the market is getting rather complicated and unless you have an astute business backer with the money to pay for expensive colour advertising, packaging and public relations, you would be well advised to leave the marketing to the professionals. Typically, a full-colour full-page advertisement in the leading computer magazines costs more than £1000!

A number of software houses, such as Llamasoft and, to a lesser extent, Ultimate only exist to release their own games written in-house. Check through the various magazines and note down those companies who:

- (a) produce quality advertising with clever promotional writing text,
- (b) advertise regularly, and
- (c) get good media coverage on the news pages, in the reviews sections, and on television programmes.

Then:

1. If you can, phone and ask who is responsible for evaluating software, and what his title is at the company.
2. Do check that your game is completely bug-free, and also remember to check for spelling mistakes in the screen displays.
3. Write a short letter saying what your game involves, detailing any unique features and indicating if any other versions of the game are available or could be produced quickly.
4. Enclose a copy of the completed game, or a portion of it, clearly marked with the name of the program, the computer used, and your name and address. Software evaluators are very busy people, so it is always useful if you can supply the software on disc. It is also helpful to send a "cheat-sheet" giving passwords or solutions to the puzzles in the game. If you have produced something very special, you could produce a video of the game in action as some software evaluators are, regrettably, untrustworthy. However, this does have its drawbacks. Sandy White sent a video of his Spectrum game Ant Attack to one particular software house who simply stuck it in a cupboard as they didn't have a video recorder at the office. Subsequently, he sent a video to Quicksilver who immediately flew Sandy down to sign him up!
5. If, after two weeks, you have heard nothing, ring the person you sent the game to and check on the game's progress through the system.

If your game is accepted:

1. Be prepared to make changes to your game. Most of these amendments (apart from any bugs discovered) should be quite straightforward to incorporate, often just meaning a change of name, new character designs or different music. Many good games have never been released simply because the programmer was not prepared to make simple improvements to the game. If you are unsure of any major changes being requested, contact the company again and discuss the matter with the software manager. Perhaps an alternative solution may be possible.
2. Find out when the the company intends to release the game and ask about their author payment structure. Don't jump to conclusions, and do listen carefully to what is being offered. To maximise your earnings, ask for a royalty for every copy sold. This way you will benefit if the game sells really well. An outright payment will often be a compromise based on the performance of the company's other games. Don't accept any percentage based on "profits" — this is too vague a term. You should really be getting about 80p per copy sold for an original title retailing at £9.95. Be prepared to accept a lower royalty if licensing is involved (to as little as 30p) — but expect much higher sales.
3. It has happened that games have been accepted, then left lying around for up to a year until the company had a clearout and rejected them. So it is in your interest to pursue the company to ensure that the game's release is going ahead at the right speed.
4. Insist on a contract and check that the written contract agrees with the verbal assurances you have been given.
5. When the game is released check the advertising and shops to ensure that the selling price is as agreed and that all the versions for other computers are available. It has been known for a programmer to be told that his game has never been released, despite the program being available in the shops. This happened with a well-known Electron program. John Dyson wrote this game and submitted it to a leading software house at their request: they published the game, but John didn't receive a single penny from them. On numerous occasions they told him that the game had never been released, but many months later a friend came across the game still on sale and sent it to him. The matter is now in the hands of John's solicitor.

The Game Concept



Many game designers are stuck in a rut at the moment with very few original concepts being developed. The market is currently swamped with commando and karate games, for example.

Game ideas can be based on your work experience, on an action sequence in a film, or around one of your hobbies. Don't blatantly rip off an arcade game. You could always use it simply as a starting point, and create something original from that seed.

Identify your target market. Making the game too strategic and less graphic will take it out of the mainstream games market. Study the top selling games on the market and ask yourself why they are doing so well. Watch out for new trends such as icons, windows and pull-down menus. Don't be afraid to ask young friends for ideas. After all, you will be wanting young people to buy your game when it is finished. They will often come up with some incredibly innovative ideas.

Once you have the basic idea for the game, you will need to decide on all the elements to go in it: the characters, the screens, the action, the puzzles.

A very useful technique is to sit down with a large note pad and write down everything that comes into your mind as you think about the concepts, but without making any attempt to evaluate their usefulness. Use a thesaurus and look up various keywords, letting your mind drift from idea to idea, noting down any words or ideas which you think may prove useful later.

Then sift through all your jottings and assemble them onto a clear sheet, linking various ideas and thus gradually developing the game design. If an idea doesn't quite fit into the game, try to think how you could change the idea so it does fit in, rather than rejecting it out of hand.

Keep a note pad near you at all times, and jot down ideas that suddenly appear to you. If you don't, you might forget the idea and the memory of even having the idea!

For further information on idea generation, consult Edward DeBono's "Lateral Thinking" (Pelican), or "Thinking Course" (Ariel), or "Use Your Head" by Tony Buzan (Ariel); all are available from good bookshops and libraries.

Writing the Game



Coding

Use a good assembler or software development kit. Document your source code well so that changes can easily be made.

Graphics

Write or use user-friendly character and map designers to make life easier for yourself.

Try to make the graphics as varied as possible by designing "graphics primitives" — small building blocks which can be combined to create bigger blocks to produce the different screens.

If you are not particularly artistic, get a friend to design the graphics for you. If your graphics designs are easy to use then your artist friend will be that much more productive.

Animation

Use as many frames as possible to animate the different characters. Four frames is really a minimum for walking in one direction. Make a dying man crumple elegantly to the floor; make him bend over and pick something up rather than have the object suddenly materialise in his hand.

Sound

As well as adding appropriate sound effects, allow room for a musical soundtrack incorporating pitch bend and portamento to give the score a human feel. Programs such as Electrosound by Orpheus for the C64 can be invaluable. But be fair and give credit where it is due on a title page.

Include sound on/off and music on/off options. Many people find sound effects distracting, and some like to play games late at night (without disturbing the neighbours).

Control

Make the game as easy to operate as possible and avoid complex key operations: the best games require very few instructions for playing the game. Don't drop the player right in the middle of the action. Ease him gently into the game so that he can get used to the controls. Not every player is an arcade ace — many are young children.

Include both joystick control and keyboard control — with user-selectable keys if possible. Some people do not have joysticks, others may prefer to use the keyboard.

Include a game pause, causing your main character to crouch, sit or tap his foot, or something completely out of the ordinary.

Finishing Touches

Include a suitable pause between each life lost and game over message so that the player

knows what has hit him, so to speak. Allow the player to abort a game easily if he is doing badly, and don't make him wait 30 seconds before he can restart because you have insisted on inserting a nifty animation routine at the start of every game.

High Score Table

High score tables are a very important feature. Players want to see that they are improving. Keep the number of entries to a maximum of 20 with sample scores added which increase in such a way that the player can see his skill developing as he slowly but surely improves his ranking in the table. The top position should be attainable but fairly difficult to achieve. Allow at least 8 characters for name input.

During game play, you could display special messages for a score of 10,000, another at 20,000 and so on. Or you could add a graphical effect: a simple change of background colour at very high levels can be immensely stimulating and rewarding, especially as the player realises that he is seeing something that very few people will ever see.

Save Game

In certain types of game a "save current position" option would be beneficial. Organise the variables so that only one or two blocks of data are saved out. If the game is sold on disc, allow the position to be stored to disc not cassette. This may seem obvious, but it is often overlooked: some disc owners do not have a tape recorder for a start.

Loading Screen

Do not skimp on this. How many times have you switched off the TV during the opening credits of a film or serial? In the same way you must stimulate and excite players by using simple animation, colour changes and music whilst the game loads in.

Game Testing

The only real way to do this is to get somebody else to play the game without your supervision, and allow him to do daft things with the game. Ask people who do not know you well, as they will be more blunt and honest about your program. You will be surprised how many bugs will be uncovered. Watch out for easier ways to solve your difficult puzzles. It may be possible to complete the game easily by finding trick ways through the game, or by solving puzzles in a different way.

Are the testers having fun? Do other people stop what they are doing and crowd round the display, or do they take a quick look then wander off? A good game should draw the crowds like the Pied Piper of Hamlyn. Note down any comments — don't rely on your memory — then retire to your computer and make improvements.

Compatibility

Check the game on different versions of your computer, and with all popular peripherals. Avoid using "illegal" techniques which, although clever, may mean that your program will not work on new versions of your computer. Check your code carefully — the top-selling game Repton contained the following code: JSR:NOP:NOP due to poor editing. This caused a JUMP to &EAEA which RTSed without any problems on the standard BBC Micro. When the same program was tried on the newly-released BBC Model B Plus the game crashed.

About Superior Software



Superior Software was set up in 1982 by Richard Hanson, who has a degree in Computational Science, and had previously written many of the Acorn Atom and early BBC titles for Program Power (now called Micro Power). Superior Software has released titles on the Oric and Commodore-64, but has preferred to concentrate its efforts on the BBC Micro and Electron. This has proved to be very rewarding both creatively and financially, and Superior Software is now regarded as the leading software house for BBC Micro and Electron software.

How a game is released

1. The Marketing Manager and Managing Director put together a marketing plan and produce a production schedule which lists all the activities which must be carried out, together with a completion date.
2. The graphics design company is briefed in full about the game and ideas for the packaging and advertising are discussed.
3. After a colour "visual" (a rough sketch) has been produced, the artwork is drawn and air-brushed. This may take 20 hours or more of concentrated work.
4. Final amendments are made to the game and any remaining bugs are removed.
5. Suitable screen shots are carefully produced. The advertising copy and game instructions are written and the advertisement and packaging are put together.
6. The advertising is booked with suitable magazines determined by careful comparison of readership surveys and circulation figures, and the cassette and disc inserts are sent for printing.
7. The distributors are visited in turn and presented with all the relevant information to enable them to decide the quantities they wish to take.
8. One computer magazine is selected and given an exclusive on the game. Press releases are written and sent out together with a review copy to all the trade and consumer magazines, as well as certain radio stations, television shows (such as Micro Live) and syndicated local newspaper columnists and freelance reviewers. Follow-up telephone calls are made to ensure that everything arrived safely.
9. The game is released and supplied to the distributors and shops. The whole process from the game being accepted to eventual release takes just 2 months. But the story does not stop there, as there are competitions to plan, special features in magazines, and a whole host of other activities still to be carried out. In the meantime, it is the turn of another game . . .

Superior Software's Titles



Superior Software has released 89 titles to date. An early release Hunchback (for the BBC Micro), was an immediate success, selling more than 30,000 copies. Recent releases have included SPEECH!, Citadel, Karate Combat, Repton and Repton 2 (for the BBC Micro and Electron).

SPEECH! has received much publicity for its innovative and impressive specification: it has been featured on television on both Micro Live and Saturday Superstore. The software is sold as a suite of 6 programs, the core being a software speech synthesiser with phoneme recognition module and English parser crammed into just 7½K of memory.

Citadel is a graphic adventure game which, due to some cleverly efficient compaction techniques, features over 100 detailed screens of action. The game took over a year of development work, and was programmed by a small team of Danish programmers.

Karate Combat, at the time of writing, stands proudly at No. 1 in the BBC Micro software charts compiled by Gallup. It is a realistic fighting simulation game and uses impressive animation techniques.

Repton is another triumph for Superior Software: a clever game idea backed with a prize competition is the basis of the success of this game. Many customers wrote to us praising the game, and time and time again we were requested to produce a follow-up program.

Repton 2, which was heralded with a series of "teaser" adverts prior to its launch, is the basic Repton theme enhanced, improved and enlarged into a mighty challenge of skill and dexterity. It was voted BBC Micro Game of the Year 1985 at the Computer Gamer awards ceremony held in January 1986.

If you need any help, or want some advice regarding any of the points mentioned in this guide, please do not hesitate to telephone or write to Richard Hanson at the address below.

Richard Hanson,
Managing Director,
Superior Software Ltd.,
Regent House,
Skinner Lane,
Leeds, LS7 1AX.
Tel: (0532) 459457

