B 1-5

FOR ANY **SINCLAIR ZX SPECTRUM**

ARCADIA the name of the game especially created to be the fastest, meanest, most addictive, shoot 'em up game you've ever desired. Wave after wave of the most loathsome and deadly aliens billow hypnotically towards your space fighter with deadly intent. But then you have dual Plasma Disruptors and an Ion Thrust Drive haven't you! ...

Made in England

This software product has a life-time guarantee If the tape should ever fail to load into your ZX SPECTRUM first time it will be replaced free of charge This guarantee will only be honoured if the faulty tape is sent directly to IMAGINE in Liverpool. If the tape has been physically damaged please include £1.50 to cover replacement costs. This guarantee is in addition to any statutory rights

GAME DESIGN by D.H. Lawson and M. Butler
SOFTWARE by D.H Lawson
GRAPHIC DESIGN/ILLUSTRATION by Steve Blower

ARCADIA will work with any ZX SPECTRUM

LOADING Place the cassette into your tape recorder with the printed side facing upwards Ensure that a lead goes from the EAR socket on the SPECTRUM to the EAR socket on the recorder and that the MIC socket is disconnected

Rewind the tape to the beginning

Type one of the following –

LOAD "arcadia"<ENTER>

or

LOAD ""<ENTER>

The word LOAD is obtained by depressing the J key and the quotes by depressing the SYMBOL SHIFT and P keys simultaneously <ENTER> means to press the key marked ENTER. Also note that there is no space between the two quotes in the second method.

Press play on the recorder

If loading is successful a flashing message will appear on the screen after several seconds The process is entirely automatic from there on.

If problems are experienced adjust your volume control and try again or re-read chapter six of your introductory manual

PLAYING ARCADIA The spaceship Arcadia has been especially equipped with dual Plasma Disruptor guns and an Ion thrust drive in order to combat the deadly menace of the many alien races of the Atarian Empire Your mission as commander of the Arcadia is quite simple – to destroy as many of the enemy as possible. Intelligence reports indicate that the Atarian fleets will attack in formation and are often quite suicidal in their methods They will attack the Arcadia in waves, each wave lasting a set period of time before it breaks off the attack If you manage to destroy the entire fleet within this time another fleet will attack you If an alien race fails to destroy the Arcadia during this period a different race will resume the attack GOOD LUCK!

THE TOP STATUS LINE.

The top row shows from left to right –
The time left before the end of the current attack wave
The number of spaceships available to you
The score so far
The number of attack levels you have survived
The highest score so far

You start with 5 lives and gain an extra one every time you survive 4 attack waves

SCORES The score for each alien destroyed is the number of the current level

HOW TO CONTROL THE SPACESHIP

There are a large number of keys which may be used You should choose the combination with which you feel most comfortable

LEFT The keys to move left are all on the lowest keyboard row Beginning with CAPS SHIFT every alternate key may be used i.e CAP SHIFT, X,V,N, SYMBOL SHIFT

RIGHT The remaining keys on the bottom row may be used to move right i.e Z,C,B,M, SPACE

THRUST Any of the keys on the second row may be used to move thrust upwards i.e A,S,D,F,G etc

FIRE Any of the keys on the third row may be used to fire your disruptor bolts i.e Q,W,E,R,T, etc In addition if a key on the top row is pressed the game will "freeze" Pressing any key not on the top row will allow you to resume the game To increase the volume of the sound effects you should connect the MIC socket on your Spectrum to the MIC socket on your cassette recorder Disconnect the ear line Remove any cassette from the recorder and press play on the recorder ARCADIA is just one of a range of exciting and original games for your ZX SPECTRUM Please see the IMAGINE advertisements in a recent computer magazine for further details

Imagine Software
Masons Buildings, Exchange Street East, Liverpool L2 3PN.

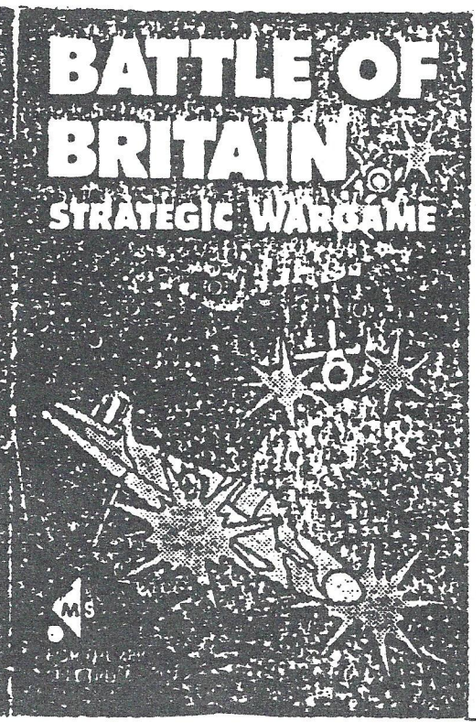
B1/7

12

~~B1~~

BATTLE OF BRITAIN

- High resolution screen map
- Variable difficulty levels
- 3 defence squadrons
- Instructions in program
- Randomly selected targets and bomber routes
- Morse sound option
- Solo game



Load name: "B" (Recorded both sides.)
Full playing instructions are on the tape.

When the game has started running the following instructions can be input by the player. Keep the key depressed until a responding message scrolls across the bottom of the map.

s: Scramble a squadron.
c: Change course to...
p: Send a squadron on preset patrol.
b: Return a squadron to base.
r: Report status of a squadron.
i: Look at the instructions.
k: Check key to commands and [illegible]
q: Quit game and restart.

Sinclair
ZX Spectrum with 48K RAM

FLIGHT Simulation

from PSION

Load and run by typing LOAD "flight"

Over the last decade, with the increasing power of computers, pilots have been trained to "fly" new airliners on large scale, computer-controlled simulators on the ground. Even on a small microcomputer like the Sinclair ZX Spectrum the essential parameters of flight, the dynamics of the airplane, the navigation of the airplane, the main instruments and the representation of the outside world can be portrayed in real-time. "FLIGHT Simulation" includes these effects and represents a small, high-performance, two-engined, propeller-driven airplane.

ASPECTS OF FLYING

The essential controls of an airplane include the joystick, flaps, rudder and engine power. Moving the joystick sideways effects the ailerons on the wings causing the plane to bank to the left or right. Moving the joystick forwards and backwards affects the elevators on the tailplane so that the nose of the airplane moves down or up respectively.
The aerodynamics of an airplane are extremely complicated. Changing one control usually has more than one effect. For example the ailerons do not simply cause the plane to roll, but produce a sideways airflow which causes the plane to turn as well. You may learn and experience these effects in the simulation.
The attitude and motion of an airplane is shown by many instruments and navigational aids in the pilot's cockpit, as discussed below. The pilot needs to use these instruments to navigate his airplane on to the right line or vector for approach to a runway, to fly his airplane on the right heading or bearing along that vector and to approach the runway with the right speed, altitude and descent angle to land the airplane. Typically, the correct approach angle of descent should be about 3° which implies an altitude of about 6,000 feet at 20 miles out, 3,000 feet at 10 miles out and 1,000 feet at just over 3 miles from the runway.
The rudder controls can contribute to the turning of the airplane. When on the ground, while taxi-ing the rudder controls also steer the direction of the airplane.

FLYING ON THE ZX SPECTRUM

Flight Simulation on the Spectrum is a full-feature program which mimics the piloting of a small airplane in real-time and in considerable detail. The detailed dynamics of an airplane are included and even looping the loop and rolling may be performed. You may land at either of two runways, take-off, navigate with the aid of beacons and in flight view the features of the world outside through the cockpit windows.
The main display is the pilot's cockpit view with a detailed instrument panel in the lower half of the screen and a view of the world outside through the cockpit windows in the top half of the screen. Through the cockpit windows you can see the horizon formed by the light sky and dark ground, the runway lights in three-dimensional perspective if you are in the vicinity of a runway, and features on the ground such as lakes, etc. As you bank, dive and climb, so the horizon and features on the ground will move accordingly through the cockpit windows.
You may switch the display however, to a navigational chart or map showing the beacons, runways and other features to help you navigate and land the plane.
After the program has been loaded from cassette, a menu will appear asking you whether you wish to take-off, start in flight or practice the final approach for landing. Press the keys 1, 2 or 3 respectively. You will be asked whether you want to include the effects of wind. Answer yes if you are skilled and can cope with the effects of wind both in landing and navigation. Otherwise press "n" for no. The program will then change immediately to the pilot's cockpit view.

THE INSTRUMENT PANEL

In the lower half of the screen in the pilot's cockpit view is the instrument panel. There are five clock-like dials, a number of gauges, warning lights and a variety of digital read-outs. The five "clocks" from left to right are the instrument landing system (ILS), the airspeed indicator, the radio-direction-finding equipment (RDF), the altimeter and the rate-of-climb indicator (ROC).
RDF clock is the large dial in the centre of the instrument panel. A small airplane is drawn in the centre of the dial and points in the direction or heading of the plane. A digital reading on the clock gives the heading in compass degrees of the airplane. The RDF is the most important navigational instrument. At any stage the plane is logged on to one of a number of beacons on the ground. The position of the current beacon at any stage relative to the direction of the airplane is represented on the RDF clock as a flashing dot near the circumference. If you wish to head directly for a beacon, bank the airplane until the flashing dot moves round the circumference to the "12 o'clock" position.
Airspeed indicator is a clock with one needle immediately to the left of the RDF. The needle points to the airspeed of the airplane measured in knots x 10.
Altimeter is a clock with two needles immediately to the right of the RDF. The small needle gives the height in units of 1000 feet and the longer needle gives the next digit as hundreds of feet.
ROC or rate-of-climb indicator is the clock-like dial on the right-hand side. It measures the vertical speed of the

airplane in units of 1000 feet per minute. When the needle points upwards above the 0, the plane is climbing and vice versa.

POWER gauge on the bottom right measures the extent of the throttle. The thrust of the engines increases with throttle but reduces in the rarer atmosphere of higher altitudes.

FUEL gauge displays the fuel remaining in the tanks.

FLAPS shows the angle of extent of the flaps. The needle points downwards with maximum flap and is horizontal with the flaps retracted.

GEAR has a green and red panel. When the undercarriage or gear is up this will be indicated in the red panel, otherwise "down" will appear in the green panel.

BCN RGE BRG is a digital readout giving information on the current logged-on beacon. BCN gives the beacon call sign of the logged-on beacon. RGE gives its range in nautical miles and BRG gives the bearing of the beacon in compass degrees relative to the airplane.

ILS is the Instrument Landing System dial on the left of the panel. It is a guidance system which aids the pilot in the approach to the runway. A radio beacon at the start of the runway emits a signal, the position of which is displayed on the ILS as a flashing dot. When the airplane is on the correct approach to the runway, the flashing dot will be at the centre of the ILS. If it is not at the centre, you the pilot should steer towards the dot. Thus if the dot, representing the runway is on the left, the pilot should bank to the left until the dot moves to the centre. If it is above the centre, the plane is too low and the joystick should be pulled back.

Ra or Radio altimeter is a digital readout and part of the ILS system. A reflected radio signal from the ground measures the height in feet of the airplane from the ground to the wheels. It gives a precise measurement for landing.

THE PILOT CONTROLS

Joy stick — The joystick of the airplane is represented on the keyboard of the ZX Spectrum by the cursor arrows (keys 5, 6, 7 and 8). Press ← (key 5) to bank left. Press → (key 8), to bank right. Press ↑ (key 7) to move the joystick forward and point the nose of the plane down for diving. Press ↓ (key 6) to pull the joystick towards you so that the nose of the plane goes up for ascent.

Rudder — The rudder on the tailplane can help to turn the plane and is controlled by the keys "Z" to turn left and "X" to turn right. While taxi-ing on the ground, the gear is steered by the rudder controls.

Power — The engines' thrust or power is controlled by the keys "P" and "O". By pressing the key "P" the throttle is increased and the engines give more power while pressing key "O" reduces the throttle and engine power (note "O" is to the left of "P").

Flaps — The extent of the flaps on the wings is controlled by the keys "F" and "D". Press the key "F" to extend the flaps further and press the key "D" to retract, or partially retract, the flaps (note key "D" is to the left of key "F"). The flaps can be extended or retracted to a varying degree (as shown on the gauge) and should only be fully extended for the final stage of landing to avoid stalling at reduced speed. With the flaps retracted, the stall speed of the plane is 80 knots, while with full flaps, the stall speed is 60 knots. Extending the flaps while the plane is at high speed could possibly damage or tear off the wings of the plane.

Gear — The gear or undercarriage can be extended by pressing the key "G". If the gear is down pressing the key "G" will retract the undercarriage. The undercarriage should not be dropped at high speed as apart from increasing the drag on the plane you may damage or destroy the undercarriage.

Beacon — To change the current logged-on beacon, press the key "B". So long as you press the key "B" the current beacon will change sequentially until you obtain the navigational beacon you require.

Map — Press the key "M" to switch the display from the cockpit pilot's display to the map or to switch back from the map to the cockpit pilot's display.

You may press several keys simultaneously.

Never press the BREAK key.

THE MAP

If you press the key "M", the display will change to a navigational chart or map showing the runways, features on the ground such as lakes and the position of the navigational beacons. The map shows the four compass points of NORTH (N-0°), EAST (E-90°), SOUTH (S-180°), WEST (W-270°). There are two airports, a large international airport called MAIN and a small local airport called CLUB. MAIN has a long runway of over a mile in length and is therefore easy to land on in a small plane. CLUB however is a small local airport and therefore has a short runway of some 800 yards. The runway of MAIN lies along the line east to west (90°-180°). Therefore on your final approach for landing the plane must be travelling on a heading of exactly 90° or 270°. The runway of CLUB on the other hand, lies along the line north to south.

The map also shows the position of the various navigational beacons and a variety of landmarks and features on the ground. Near the airport MAIN, there are two beacons some three miles beyond each end of the runway with call signs of ME and MW. Airport CLUB has two beacons CN and CS two miles beyond each end of the runway. There are three other navigational beacons OA OB and OC.

NAVIGATION

The most difficult part of flying is the approach and landing at an airfield. You can experiment with the controls in varying the speed, altitude and direction of the plane at a reasonably high altitude without worrying about the navigation. If you wish to land the plane, however, you will have to navigate the plane on to the right vector and on to the right course, and you will have to approach the runway at roughly the right glide angle. This is a difficult task and requires a lot of practice and experience before you can achieve a landing successfully.

The map and instruments will help you to determine your position precisely. You will then need to think of the approximate manoeuvres and course to approach the airfield on the right flight path. The runway of airport MAIN lies east to west in the simulation on a line (or vector) from 90° to 270° or vice-versa. You may approach the runway from either end. For example, if you wish to approach the runway from the east to the west, you must first manoeuvre the airplane far to the east of the runway. If you use the beacon ME you will need to fly the plane until the beacon is on a bearing (or vector) of 270°. If you then bank on to this bearing on a course or heading of 270° you will be flying on exactly the right flight path for your approach to the runway. As you fly towards the beacon ME, to keep on the right flight path you must ensure that both the heading and the beacon coincide at 270°. As you fly over the beacon in the later stages of the approach, the bearing of the beacon will of course change to 90°. Similarly, you may use any of the other beacons to set a course for a particular flight objective. When heading directly for a beacon, remember that your heading and the beacon bearing must always coincide precisely. Flying is difficult for the uninitiated and if you have trouble navigating yourself to the runway for final landing you can always use the option at the beginning of the program to give you an automatic approach and allow you to experiment with the final touchdown.

Once you have touched down, you must reduce the power to zero to bring the plane to a halt. You may taxi and steer using the rudder controls and take off again.

Key	Pilot Control
↑	Joystick forward (dive)
↓	Joystick backward (climb)
→	Joystick to right (bank to right)
←	Joystick to left (bank to left)
Z	Rudder control (left)
X	Rudder control (right)
P	Increase throttle (more power)
O	Decrease throttle (less power)
F	Increase flap extent
D	Decrease flap extent
G	Lower gear (undercarriage) if up. Raise gear (undercarriage) if down.
B	Change beacon sequentially
M	Switch from cockpit display to navigational Map or back to cockpit.

Sinclair ZX Spectrum with 48K RAM

FLIGHT Simulation

from PSION

Load and run by typing LOAD "flight"

Over the last decade, with the increasing power of computers, pilots have been trained to "fly" new airliners on large scale, computer-controlled simulators on the ground. Even on a small microcomputer like the Sinclair ZX Spectrum the essential parameters of flight, the dynamics of the airplane, the navigation of the airplane, the main instruments and the representation of the outside world can be portrayed in real-time. "FLIGHT Simulation" includes these effects and represents a small, high-performance, two-engined, propeller-driven airplane.

ASPECTS OF FLYING

The essential controls of an airplane include the joystick, flaps, rudder and engine power. Moving the joystick sideways affects the ailerons on the wings causing the plane to bank to the left or right. Moving the joystick forwards and backwards affects the elevators on the tailplane so that the nose of the airplane moves down or up respectively.
The aerodynamics of an airplane are extremely complicated. Changing one control usually has more than one effect. For example the ailerons do not simply cause the plane to roll, but produce a sideways airflow which causes the plane to turn as well. You may learn and experience these effects in the simulation.
The attitude and motion of an airplane is shown by many instruments and navigational aids in the pilot's cockpit, as discussed below. The pilot needs to use these instruments to navigate his airplane on to the right line or vector for approach to a runway, to fly his airplane on the right heading or bearing along that vector and to approach the runway with the right speed, altitude and descent angle to land the airplane. Typically, the correct approach angle of descent should be about 3° which implies an altitude of about 6,000 feet at 20 miles out, 3,000 feet at 10 miles out and 1,000 feet at just over 3 miles from the runway.
The rudder controls can contribute to the turning of the airplane. When on the ground, while taxi-ing the rudder controls also steer the direction of the airplane.

FLYING ON THE ZX SPECTRUM

Flight Simulation on the Spectrum is a full-feature program which mimics the piloting of a small airplane in real-time and in considerable detail. The detailed dynamics of an airplane are included and even looping the loop and rolling may be performed. You may land at either of two runways, take-off, navigate with the aid of beacons and in flight view the features of the world outside through the cockpit windows.
The main display is the pilot's cockpit view with a detailed instrument panel in the lower half of the screen and a view of the world outside through the cockpit windows in the top half of the screen. Through the cockpit windows you can see the horizon formed by the light sky and dark ground, the runway lights in three-dimensional perspective if you are in the vicinity of a runway, and features on the ground such as lakes, etc. As you bank, dive and climb, so the horizon and features on the ground will move accordingly through the cockpit windows.
You may switch the display however, to a navigational chart or map showing the beacons, runways and other features to help you navigate and land the plane.
After the program has been loaded from cassette, a menu will appear asking you whether you wish to take off, start in flight or practice the final approach for landing. Press the keys 1, 2 or 3 respectively. You will be asked whether you want to include the effects of wind. Answer yes if you are skilled and can cope with the effects of wind both in landing and navigation. Otherwise press "n" for no. The program will then change immediately to the pilot's cockpit view.

THE INSTRUMENT PANEL

In the lower half of the screen in the pilot's cockpit view is the instrument panel. There are five clock-like dials, a number of gauges, warning lights and a variety of digital read-outs. The five "clocks" from left to right are the instrument landing system (ILS), the airspeed indicator, the radio-direction-finding equipment (RDF), the altimeter and the rate-of-climb indicator (ROC).
RDF clock is the large dial in the centre of the instrument panel. A small airplane is drawn in the centre of the dial and points in the direction or heading of the plane. A digital reading on the clock gives the heading in compass degrees of the airplane. The RDF is the most important navigational instrument. At any stage the plane is logged on to one of a number of beacons on the ground. The position of the current beacon at any stage relative to the direction of the airplane is represented on the RDF clock as a flashing dot near the circumference. If you wish to head directly for a beacon, bank the airplane until the flashing dot moves round the circumference to the "12 o'clock" position.
Airspeed Indicator is a clock with one needle immediately to the left of the RDF. The needle points to the airspeed of the airplane measured in knots x 10.
Altimeter is a clock with two needles immediately to the right of the RDF. The small needle gives the height in units of 1000 feet and the longer needle gives the hundreds of feet.
ROC or rate-of-climb indicator is the clock-like dial on the right-hand side. It measures the vertical speed of the airplane in units of 1000 feet per minute. When the needle points upwards above the 0, the plane is climbing and vice versa.

POWER gauge on the bottom right measures the extent of the throttle. The thrust of the engines increases with throttle but reduces in the rarer atmosphere of higher altitudes.

FUEL gauge displays the fuel remaining in the tanks.

FLAPS shows the angle of extent of the flaps. The needle points downwards with maximum flap and is horizontal with the flaps retracted.

GEAR has a green and red panel. When the undercarriage or gear is up this will be indicated in the red panel, otherwise "down" will appear in the green panel.

BCN RGE BRG is a digital readout giving information on the current logged-on beacon. BCN gives the beacon call sign of the logged-on beacon. RGE gives its range in nautical miles and BRG gives the bearing of the beacon in compass degrees relative to the airplane.

ILS is the Instrument Landing System dial on the left of the panel. It is a guidance system which aids the pilot in the approach to the runway. A radio beacon at the start of the runway emits a signal, the position of which is displayed on the ILS as a flashing dot. When the airplane is on the correct approach to the runway, the flashing dot will be at the centre of the ILS. If it is not at the centre, you the pilot should steer towards the dot. Thus if the dot, representing the runway is on the left, the pilot should bank to the left until the dot moves to the centre. If it is above the centre, the plane is too low and the joystick should be pulled back.

Ra or **Radio altimeter** is a digital readout and part of the ILS system. A reflected radio signal from the ground measures the height in feet of the airplane from the ground to the wheels. It gives a precise measurement for landing.

THE PILOT CONTROLS

Joy stick — The joystick of the airplane is represented on the keyboard of the ZX Spectrum by the cursor arrows (keys 5, 6, 7 and 8). Press ← (key 5) to bank left. Press → (key 8), to bank right. Press ↑ (key 7) to move the joystick forward and point the nose of the plane down for diving. Press ↓ (key 6) to pull the joystick towards you so that the nose of the plane goes up for ascent.
Rudder — The rudder on the tailplane can help to turn the plane and is controlled by the keys "Z" to turn left and "X" to turn right. While taxi-ing on the ground, the gear is steered by the rudder controls.
Power — The engines' thrust or power is controlled by the keys "P" and "O". By pressing the key "P" the throttle is increased and the engines give more power while pressing key "O" reduces the throttle and engine power (note "O" is to the left of "P").
Flaps — The extent of the flaps on the wings is controlled by the keys "F" and "D". Press the key "F" to extend the flaps further and press the key "D" to retract, or partially retract, the flaps (note key "D" is to the left of key "F"). The flaps can be extended or retracted to a varying degree (as shown on the gauge) and should only be fully extended for the final stage of landing to avoid stalling at reduced speed. With the flaps retracted, the stall speed of the plane is 80 knots, while with full flaps, the stall speed is 60 knots. Extending the flaps while the plane is at high speed could possibly damage or tear off the wings of the plane.
Gear — The gear or undercarriage can be extended by pressing the key "G". If the gear is down pressing the key "G" will retract the undercarriage. The undercarriage should not be dropped at high speed as apart from increasing the drag on the plane you may damage or destroy the undercarriage.
Beacon — To change the current logged-on beacon, press the key "B". So long as you press the key "B" the current beacon will change sequentially until you obtain the navigational beacon you require.
Map — Press the key "M" to switch the display from the cockpit pilot's display to the map or to switch back from the map to the cockpit pilot's display.
You may press several keys simultaneously.
Never press the BREAK key.

THE MAP

If you press the key "M", the display will change to a navigational chart or map showing the runways, features on the ground such as lakes and the position of the navigational beacons. The map shows the four compass points of NORTH (N-0°), EAST (E-90°), SOUTH (S-180°), WEST (W-270°). There are two airports, a large international airport called MAIN and a small local airport called CLUB. MAIN has a long runway of over a mile in length and is therefore easy to land on in a small plane. CLUB however is a small local airport and therefore has a short runway of some 800 yards.
The runway of MAIN lies along the line east to west (90°-180°). Therefore on your final approach for landing the plane must be travelling on a heading of exactly 90° or 270°. The runway of CLUB on the other hand, lies along the line north to south.
The map also shows the position of the various navigational beacons and a variety of landmarks and features on the ground. Near the airport MAIN, there are two beacons some three miles beyond each end of the runway with call signs of ME and MW. Airport CLUB has two beacons CN and CS two miles beyond each end of the runway. There are three other navigational beacons OA OB and OC.

NAVIGATION

The most difficult part of flying is the approach and landing at an airfield. You can experiment with the controls in varying the speed, altitude and direction of the plane at a reasonably high altitude without worrying about the navigation. If you wish to land the plane, however, you will have to navigate the plane on to the right vector and on to the right course, and you will have to approach the runway at roughly the right glide angle. This is a difficult task and requires a lot of practice and experience before you can achieve a landing successfully.
The map and instruments will help you to determine your position precisely. You will then need to think of the approximate manoeuvres and course to approach the airfield on the right flight path. The runway of airport MAIN lies east to west in the simulation on a line (or vector) from 90° to 270° or vice versa. You may approach the runway from either end. For example, if you wish to approach the runway from the east to the west, you must first manoeuvre the airplane far to the east of the runway. If you use the beacon ME you will need to fly the plane until the beacon is on a bearing (or vector) of 270°. If you then bank on to this bearing on a course or heading of 270° you will be flying on exactly the right flight path for your approach to the runway. As you fly towards the beacon ME, to keep on the right flight path you must ensure that both the heading and the beacon coincide at 270°. As you fly over the beacon in the later stages of the approach, the bearing of the beacon will of course change to 90°. Similarly, you may use any of the other beacons to set a course for a particular flight objective. When heading directly for a beacon, remember that your heading and the beacon bearing must always coincide precisely. Flying is difficult for the uninitiated and if you have trouble navigating yourself to the runway for final landing you can always use the option at the beginning of the program to give you an automatic approach and allow you to experiment with the final touchdown.
Once you have touched down, you must reduce the power to zero to bring the plane to a halt. You may taxi and steer using the rudder controls and take-off again.

Key	Pilot Control
↑	Joystick forward (dive)
↓	Joystick backward (climb)
→	Joystick to right (bank to right)
←	Joystick to left (bank to left)
Z	Rudder control (left)
X	Rudder control (right)
P	Increase throttle (more power)
O	Decrease throttle (less power)
F	Increase flap extent
D	Decrease flap extent
G	Lower gear (undercarriage) if up. Raise gear (undercarriage) if down.
B	Change beacon sequentially
M	Switch from cockpit display to navigational Map or back to cockpit

Made in [illegible]

Sinclair
ZX Spectrum with 48K RAM

FLIGHT Simulation

from PSION

Load and run by typing LOAD "flight"

Over the last decade, with the increasing power of computers, pilots have been trained to "fly" new airliners on large scale, computer-controlled simulators on the ground. Even on a small microcomputer like the Sinclair ZX Spectrum the essential parameters of flight, the dynamics of the airplane, the navigation of the airplane, the main instruments and the representation of the outside world can be portrayed in real-time. "FLIGHT Simulation" includes these effects and represents a small, high-performance, two-engined, propeller-driven airplane.

ASPECTS OF FLYING

The essential controls of an airplane include the joystick, flaps, rudder and engine power. Moving the joystick sideways affects the ailerons on the wings causing the plane to bank to the left or right. Moving the joystick forwards and backwards affects the elevators on the tailplane so that the nose of the airplane moves down or up respectively.

The aerodynamics of an airplane are extremely complicated. Changing one control usually has more than one effect. For example the ailerons do not simply cause the plane to roll, but produce a sideways airflow which causes the plane to turn as well. You may learn and experience these effects in the simulation.

The attitude and motion of an airplane is shown by many instruments and navigational aids in the pilot's

cockpit, as discussed below. The pilot needs to use these instruments to navigate his airplane on to the right line or vector for approach to a runway, to fly his airplane on the right heading or bearing along that vector and to approach the runway with the right speed, altitude and descent angle to land the airplane. Typically, the correct approach angle of descent should be about 3° which implies an altitude of about 6,000 feet at 20 miles out, 3,000 feet at 10 miles out and 1,000 feet at just over 3 miles from the runway.

The rudder controls can contribute to the turning of the airplane. When on the ground, while taxi-ing the rudder controls also steer the direction of the airplane.

FLYING ON THE ZX SPECTRUM

Flight Simulation on the Spectrum is a full-feature program which mimics the piloting of a small airplane in real-time and in considerable detail. The detailed dynamics of an airplane are included and even looping the loop and rolling may be performed. You may land at either of two runways, take-off, navigate with the aid of beacons and in flight view the features of the world outside through the cockpit windows.

The main display is the pilot's cockpit view with a detailed instrument panel in the lower half of the screen and a view of the world outside through the cockpit windows in the top half of the screen. Through the cockpit windows you can see the horizon formed by the light sky and dark ground, the runway lights in three-dimensional perspective if you are in the vicinity of a runway, and features on the ground such as lakes, etc. As you bank, dive and climb, so the horizon and features on the ground will move accordingly through the cockpit windows.

You may switch the display however, to a navigational chart or map showing the beacons, runways and other features to help you navigate and land the plane.

After the program has been loaded from cassette, a

menu will appear asking you whether you wish to t off, start in flight or practice the final approach for landing. Press the keys 1, 2 or 3 respectively. You w asked whether you want to include the effects of w Answer yes if you are skilled and can cope with the effects of wind both in landing and navigation Otherwise press "n" for no. The program will then change immediately to the pilot's cockpit view.

THE INSTRUMENT PANEL

In the lower half of the screen in the pilot's cockpit is the instrument panel. There are five clock like dia number of gauges, warning lights and a variety of d read-outs. The five "clocks" from left to right are th instrument landing system (ILS), the airspeed indic the radio-direction-finding equipment (RDF), the altimeter and the rate-of-climb indicator (ROC).

RDF clock is the large dial in the centre of the instrument panel. A small airplane is drawn in the c of the dial and points in the direction or heading of plane. A digital reading on the clock gives the headi in compass degrees of the airplane. The RDF is the most important navigational instrument. At any ti the plane is logged on to one of a number of beaco the ground. The position of the current beacon at a stage relative to the direction of the airplane is represented on the RDF clock as a flashing dot ne circumference. If you wish to head directly for a beacon, bank the airplane until the flashing dot mo round the circumference to the "12 o'clock" positi

Airspeed Indicator is a clock with one needle immediately to the left of the RDF. The needle poi the airspeed of the airplane measured in knots x 10

Altimeter is a clock with two needles immediately the right of the RDF. The small needle gives the he in units of 1000 feet and the longer needle gives th digit as hundreds of feet.

ROC or rate of climb indicator is the clock like dia the right-hand side. It measures the vertical speed

airplane in units of 1000 feet per minute. When the needle points upwards above the 0, the plane is climbing and vice versa.

POWER gauge on the bottom right measures the extent of the throttle. The thrust of the engines increases with throttle but reduces in the rarer atmosphere of higher altitudes.

FUEL gauge displays the fuel remaining in the tanks.

FLAPS shows the angle of extent of the flaps. The needle points downwards with maximum flap and is horizontal with the flaps retracted.

GEAR has a green and red panel. When the undercarriage or gear is up this will be indicated in the red panel, otherwise "down" will appear in the green panel.

RCN RGE BRG is a digital readout giving information on the current logged-on beacon. RCN gives the beacon call sign of the logged-on beacon. RGE gives its range in nautical miles and BRG gives the bearing of the beacon in compass degrees relative to the airplane.

ILS is the Instrument Landing System dial on the left of the panel. It is a guidance system which aids the pilot in the approach to the runway. A radio beacon at the start of the runway emits a signal, the position of which is displayed on the ILS as a flashing dot. When the airplane is on the correct approach to the runway, the flashing dot will be at the centre of the ILS. If it is not at the centre, you the pilot should steer towards the dot. Thus if the dot, representing the runway is on the left, the pilot should bank to the left until the dot moves to the centre. If it is above the centre, the plane is too low and the joystick should be pulled back.

Rad or Radio altimeter is a digital readout and part of the ILS system. A reflected radio signal from the ground measures the height in feet of the airplane from the ground to the wheels. It gives a precise measurement during landing

THE PILOT CONTROLS

Joy stick — The joystick of the airplane is represented on the keyboard of the ZX Spectrum by the cursor arrows (keys 5, 6, 7 and 8). Press ← (key 5) to bank left. Press → (key 8), to bank right. Press ↑ (key 7) to move the joystick forward and point the nose of the plane down for diving. Press ↓ (key 6) to pull the joystick towards you so that the nose of the plane goes up for ascent.

Rudder — The rudder on the tailplane can help to turn the plane and is controlled by the keys "Z" to turn left and "X" to turn right. While taxi-ing on the ground, the gear is steered by the rudder controls.

Power — The engines' thrust or power is controlled by the keys "P" and "O". By pressing the key "P" the throttle is increased and the engines give more power while pressing key "O" reduces the throttle and engine power (note "O" is to the left of "P").

Flaps — The extent of the flaps on the wings is controlled by the keys "F" and "D". Press the key "F" to extend the flaps further and press the key "D" to retract, or partially retract, the flaps (note key "D" is to the left of key "F"). The flaps can be extended or retracted to a varying degree (as shown on the gauge) and should only be fully extended for the final stage of landing to avoid stalling at reduced speed. With the flaps retracted, the stall speed of the plane is 80 knots, while with full flaps, the stall speed is 60 knots. Extending the flaps while the plane is at high speed could possibly damage or tear off the wings of the plane.

Gear — The gear or undercarriage can be extended by pressing the key "G". If the gear is down pressing the key "G" will retract the undercarriage. The undercarriage should not be dropped at high speed as apart from increasing the drag on the plane you may damage or destroy the undercarriage.

Beacon — To change the current logged on beacon,

press the key "B". So long as you press the key "B" the current beacon will change sequentially until you obtain the navigational beacon you require.

Map — Press the key "M" to switch the display from the cockpit pilot's display to the map or to switch back from the map to the cockpit pilot's display.

You may press several keys simultaneously.

Never press the BREAK key.

THE MAP

If you press the key "M", the display will change to a navigational chart or map showing the runways, features on the ground such as lakes and the position of the navigational beacons. The map shows the four compass points of NORTH (N-0°), EAST (E-90°), SOUTH (S-180°), WEST (W-270°). There are two airports, a large international airport called MAIN and a small local airport called CLUB. MAIN has a long runway of over a mile in length and is therefore easy to land on in a small plane. CLUB however is a small local airport and therefore has a short runway of some 800 yards.

The runway of MAIN lies along the line east to west (90°-180°). Therefore on your final approach for landing the plane must be travelling on a heading of exactly 90° or 270°. The runway of CLUB on the other hand, lies along the line north to south.

The map also shows the position of the various navigational beacons and a variety of landmarks and features on the ground. Near the airport MAIN, there are two beacons some three miles beyond each end of the runway with call signs of ME and MW. Airport CLUB has two beacons CN and CS two miles beyond each end of the runway. There are three other navigational beacons OA OB and OC.

NAVIGATION

The most difficult part of flying is the approach and landing at an airfield. You can experiment with the controls in varying the speed, altitude and direction of

the plane at a reasonably high altitude without worrying about the navigation. If you wish to land the plane, however, you will have to navigate the plane on to the right vector and on to the right course, and you will have to approach the runway at roughly the right glide angle. This is a difficult task and requires a lot of practice and experience before you can achieve a landing successfully.

The map and instruments will help you to determine your position precisely. You will then need to think of the approximate manoeuvres and course to approach the airfield on the right flight path. The runway of airport MAIN lies east to west in the simulation on a line (or vector) from 90° to 270° or vice versa. You may approach the runway from either end. For example, if you wish to approach the runway from the east to the west, you must first manoeuvre the airplane far to the east of the runway. If you use the beacon ME you will need to fly the plane until the beacon is on a bearing (or vector) of 270°. If you then bank on to this bearing on a course or heading of 270° you will be flying on exactly the right flight path for your approach to the runway. As you fly towards the beacon ME, to keep on the right flight path you must ensure that both the heading and the beacon coincide at 270°. As you fly over the beacon in the later stages of the approach, the bearing of the beacon will of course change to 90°. Similarly, you may use any of the other beacons to set a course for a particular flight objective. When heading directly for a beacon, remember that your heading and the beacon bearing must always coincide precisely. Flying is difficult for the uninitiated and if you have trouble navigating yourself to the runway for final landing you can always use the option at the beginning of the program to give you an automatic approach and allow you to experiment with the final touchdown.

Once you have touched down, you must reduce the power to zero to bring the plane to a halt. You may taxi and steer using the rudder controls and take off again

Key	Pilot Control
↑	Joystick forward (dive)
↓	Joystick backward (climb)
→	Joystick to right (bank to right)
←	Joystick to left (bank to left)
Z	Rudder control (left)
X	Rudder control (right)
P	Increase throttle (more power)
O	Decrease throttle (less power)
F	Increase flap extent
D	Decrease flap extent
G	Lower gear (undercarriage) if up. Raise gear (undercarriage) if down
B	Change beacon sequentially
M	Switch from cockpit display to navigational Map or back to cockpit

Made in UK

I

SUPER C AND COPYRIGHT OF COMPILED PROGRAMS

The question of copyright, at the time of writing at least, is rather unclear in the field of computer software. However, it is generally acknowledged that ideas and actual programs can both be copyrighted. Thus if you create a program with this compiler which you wish to market commercially then you will of course have the rights to the ideas etc behind the program. However, the actual code will have been created by SUPER C at a saving of typically many months work over the alternative of your writing the machine code yourself. Most important , though, all compiled programs must contain within them SUPER C's run-time routines which remain our copyright. It is for this reason that we retain interest in the final product as a commercial program.

Hence, PLEASE CONTACT US IF YOU INTEND TO MARKET ANY PROGRAM COMPILED WITH SUPER C.

We are very interested in considering any compiled program to market ourselves - and we can offer you the best royalties and distribution of any software house - but we would also consider a small royalty should you wish to market the product yourself.

IMPORTANT!!

Whilst SUPER C is extremely easy to use when you consider the task it performs, please DO READ THIS MANUAL CAREFULLY. If you run up against any problems we will endevor to answer them, but please do write to us rather than telephoning - we do tend to get rather a lot of interest in SUPER C !

HINTS/TIPS/BUGS

Naturally, if there were any bugs to be found we think we've already got rid of them - but if you should happen to find one do tell us! We'd also love to hear what you've acheived with this superb program, so send us your results, hints/tips etc and we'll start up a newsletter for compiler owners!

INTRODUCTION

This SUPER C 48K ZX Spectrum Compiler is possibly the most sophisticated piece of software in your library. We feel sure that you will share our view that it radically transforms the Spectrum into the fastest easiest to use micro for under £200. Whilst we are sure that you will be impressed with this compiler's capabilities, please do not hesitate to contact us with any queries.

What is a compiler?

Essentially, a compiler is a program which converts a program written in BASIC into a machine code program which does the same job. After compilation there is no BASIC left in memory, and instead you have a super-fast version of the program you have devised which can be SAVEd to tape just as any other machine-code program.

SUPER C converts almost any Sinclair BASIC program into machine code with the exception of statements involving floating point arithmetic and string handling/arrays. I will explain later how you may handle strings and arrays even though the Sinclair BASIC statements are not recognised, and you may like to note that an enhancement package is available separately which upgrades this compiler to enable it to handle these few missing features.

The SUPER C Compiler sits just above the 32K mark in RAM and uses effectively some 20K of RAM for the code which makes it up and space to carry out the compilation. You are thus left with as much program memory as on a 16K Spectrum with a further 10K of RAM, or so, available above the compiler for the storage of data, strings, and so forth.

SUPER C was designed to be as user friendly as the Spectrum itself, and you should find it remarkably easy to create fast, professional machine code programs with it.

LOADING THE COMPILER

To load SUPER C simply type in LOAD"" and press <ENTER>. Leave the tape running until a message appears at the bottom of the screen telling you that the loading is complete. You will see a BASIC loader program load first, followed closely by two blocks of machine-code. Once the loading is complete you will find that

there is now just one long program line in memory. To erase this you simply type in the line number followed by pressing < ENTER >. You are now ready to type in or load in from tape/microdrive your BASIC program to be compiled.

By way of illustration we have included three demonstration programs on the tape, and you might like to run these first to get some idea of how the compiler works and just how impressive results with it can be.

LOAD the first demo by simply typing LOAD"" <ENTER> and RUN it as a BASIC program first to see how fast (slow?) it is. That done simply type in this line which will cause the BASIC to be compiled in a fraction of a second:

RANDOMISE USR 49152 <ENTER>

Almost instantly the standard Sinclair 'OK' report should have appeared at the bottom of the screen. By the way, this is probably a good point to note that although the compiler has changed the screen and boarder to blue and the ink to white, you may change them to any other colour you like. When a program is first compiled the compiler will change it back to blue and white again, but this does not effect the colours in your program.

If the demonstration compiled correctly you should find that pressing <ENTER> again just reveals a single line of BASIC:

1 PROPERTY OF SOFTEK
ALL COMMERCIAL RIGHTS RESERVED

Now simply press RUN and <ENTER> to see the same program again, but this time quite a bit faster! (N Statement Lost is a normal SUPER C report)

Do the same with DEMO 2 which counts from 0 to 1000. Without printing to the screen BASIC takes several seconds, whereas the compiled version is almost instant. Even when, by adding PRINT A, printing to the screen occurs, the compiled version is still many times faster (try it).

The third demonstration is possibly the most persuasive as it is a simple 'invader' type game which is almost unplayable in BASIC but quite a formidable game once compiled. Note how smooth the base ship's movements are! (Note: CLEAR 32000 before LOADing this!)

IS THAT ALL THERE IS TOO IT?

Essentially, yes. Every program you write can be compiled equally simply and you will only have to look out for a few 'quirks' which the compiler requires you to take note of.

PLEASE DO READ THIS MANUAL CAREFULLY!

Good luck.
Tim Langdell,
London 1984

FIRST STEPS IN USING THE COMPILER

Why has the paper and border colour been changed to blue and the ink to white? We made SUPER C do this for no other reason than after some experimentation we felt this colour combination to be the most restful to the eye. You may of course change this to any other combination. You will also note that upon loading you are immediately in the CAPS mode. This has been done because SUPER C will only recognise variables entered as capital letters, although using lower case in PRINT statements is fine.

Why was I able to just type RUN to run the DEMOs?

We have done our best to make this compiler as easy to use as possible, especially for the user unexperienced in machine-code. For this reason we made the compiled code easy to run by treating it almost like a BASIC program. You hence just press <RUN> and <ENTER> . The machine code is in fact run by a USR call in line 4 (the only other BASIC line in memory) which we've hidden behind attribute changes. You can call a LISTing of line 4 and see that it says:

4 RANDOMISE USR 25394

This gives you the start address of the resultant machine-code should you wish to add a BASIC program too and call the compiled code from it. But beware, you only have about 256 bytes of RAM left in which a BASIC program can reside with the compiled code.

What happens to the BASIC not compiled?

This is a question we are frequently asked. In fact there never is any BASIC which is not compiled. This is because the compiler looks through your BASIC program once and refuses to compile it if any 'illegal' statements are present. Thus when it compiles a program it will change all of it into machine code. If you want to mix machine-code and BASIC then as mentioned above you can add up to 256 bytes worth. You can go from BASIC to code to BASIC again, but we advise caution - see later.

What BASIC will SUPER C not compile?

SUPER C cannot handle floating point numbers and it cannot handle strings and arrays in the normal manner that Spectrum BASIC does. It can, though, of course PRINT strings to the screen and so on.

Virtually all programs without these features will compile first time, but SUPER C does have a few quirk s and there are a handful of error reports which can occur for one of a few minor reasons:

0,1	OK - compiled fine
C,X	Illegal command in line number X
6,X	GOTO or GOSUB wrong - where X will usually be the line number you were sending it to.

While running:

N,X	Run over the end of program in line X

You may also get:

A,X	Invalid argument - if you try to use some Spectrum features which SUPER C cannot handle.

In a similar vein, you may get a report, RAMTOP NO GOOD for similar reasons - this rarely means that RAMTOP is no good!

One of the most common error messages you will encounter is a:

4: Out of memory error

and this is most likely to happen when you've just compiled one program and try to load another in from tape. This is due to the compiler puting the prior code in the BASIC program area - recall the USR address is 25394. Thus to load another program type in

CLEAR 32000 ⟨ENTER⟩

Then load your next program, or type it in from the keyboard.

Look carefully through the following pages at the do's and do not's of SUPER C usage, and try to always include a line with REM B in it so that your compiled program can be stopped if it is not doing what you had hoped. REM B, then, tests to see if the BREAK (SPACE) key is pressed - else you'll never break out of the compiled program!

The other report code which may have been puzzling and which you have met already is:

N Statement lost, 4:1

This is the compiler's way of saying all went all right and it has returned to BASIC. This report will occur whenever the compiled version returns to BASIC except when you actually tell it to return by compiling a line with RETURN in it (this keyword acts as a GOSUB return, but when no GOSUB was encountered the compiler interprets it as a return to BASIC).

What can SUPER C handle?

It can deal with: CHR$,IN,PEEK,POINT,RND,USR,BEEP,BORDER, BRIGHT,CIRCLE,CLS,COPY,DATA,DRAW,FLASH,FOR-TO-NEXT, STEP,GOTO,GOSUB,IF-THEN,RETURN,INK,PAPER,OVER,INVERSE, LET,NEW,OUT,PAUSE,PLOT,POKE,PRINT(AT),RANDOMISE,READ, DATA, RESTORE, RUN,ATTR,CODE SCREEN$, STOP,+ / - *.

It uses CODE INKEY$ and CODE SCREEN$ and has 3 REM statements that it recognises as enhancements to Sinclair BASIC:

REM B - test to see if the BREAK/SPACE key pressed

REM M,x,y,z,... - When the compiler sees a REM M statement it takes the numbers after the M to be decimal bytes of machine code to be put into memory as they are, and to to be executed at that point in the program.

REM N - This executes a total NEW of the system, wiping SUPER C from memory without 'pulling the plug'.

In brief, there's no program than cannot be compiled. You can add your own customised machine code routines if you wish, make a program possible, or impossible to break out of, and with thought even compile floating point calculations - or handle them by returns to BASIC occasionally. See later.

QUIRKS OF COMPILER BASIC

In the vast majority of cases you will be able to type in your BASIC just as you have ever done. However, there are some restrictions, most of which relate to the absence of string handling or floating point arithmetic.

VARIABLES

You have 26 variables A to Z all of which must be in capital letters. Should you put in a lower case variable by mistake, attempting to compile the BASIC (or source code as its known in this context) will result in 'Nonsense in BASIC', or 'Invalid argument'.

Passing values of variables from BASIC to the COMPILER, or vice versa is possible but requires some thought. Essentially, because the compiler will only handle integers (without the enhancement, that is) you will have to ensure that you are passing integers between the programs. Any given BASIC variable,then, from 0 to 65536 would need to be divided by 256 and the integer part POKEd into a loction in high memory (54000 onwards). This is the most significant byte, and the least significant one is POKEd into the next high-memory location. A an example, lets pass variable A in BASIC to the COMPILER, and say it has a value of 2057. Thus the most significant byte will be:

INT(2057/256) =8

and the least significant byte:

2057-(8*256) = 9

That is 2057 =(8*256)+ 9

And in the program you compile you will need to PEEK the locations you have chosen to store the 8 and 9 in. Say 54000 & 54001. Thus you would need a line like this:

```
100 LET X=PEEK 54000:LET X=X*256:LET A=X+9
```

And the COMPILER'S variable A will now have the same value as the BASIC's.

To make matters easier for you we have devised a short machine code program which you can put as part of your BASIC program to be compiled (in a REM M statement - see later) which swops the variable 'A' in the compiler with the first variable in the BASIC variable table. Note that this is not necessarily the variable 'A' in BASIC - it is the first variable defined in your program accompanying the compiled program. Thus if the lowest line number in the 256 bytes maximum of BASIC accompanying the machine code is, say, LET T=0, then this routine will swop the value of T with the value of variable A in the compiler. IMPORTANT: please remember that you can only deal with integers, and there is no error checking should you try to swop a floating point number in BASIC with the compiler's 'A'.

The HEX of the routine is:

```
ED 5B 00 60 2A 4B 5C 23 23 23 4E 23 46 72
2B 73 ED 43 00 60
```

(20 bytes)

The disassembly (using our Sofmon™):

(There is no significance in the choice of location 7000H!)

```
:Disassemble 7000
7000 ED5B0060   LD   DE,(6000)
7004 2A4B5C     LD   HL,(5C4B)
7007 23         INC  HL
7008 23         INC  HL
7009 23         INC  HL
700A 4E         LD   C,(HL)
700B 23         INC  HL
700C 46         LD   B,(HL)
700D 72         LD   (HL),D
700E 2B         DEC  HL
700F 73         LD   (HL),E
7010 ED430060   LD   (6000),BC
```

And what you actually type in to the program to be compiled at the point where this is to take place is:

```
100 REM M,237,91,00,96,42,75,92,35,35,35,78,
```

NEW

As soon as at least one program has been compiled, then pressing ⟨NEW⟩ will clear memory only up to the compiler - it itself will remain ready for the next compilation. This would be true too if prior to loading another program you were to execute the following sequence:

CLEAR 32000: ⟨NEW⟩ ⟨ENTER⟩

You can, though take the option with SUPER C to compile the following statement , which when RUN will execute a total NEW of the system much like that which occurs upon power-up of your Spectrum:

1 REM N (followed by RANDOMISE USR 49152, and ⟨RUN⟩ of course)

INKEY$

This is much the same as in usual BASIC except that the lack of string handling means you will have to use either CODE INKEY$ or CHR$ CODE INKEY$. However, these can be used in almost all usual circumstances, such as POKE nnnn, CODE INKEY$ or PRINT CHR$ CODE INKEY$, etc.

PRINT

Once again this is much the same as usual. You can PRINT AT, PRINT CHR$, but you cannot PRINT TAB. Instead you will have to use PRINT AT.

FOR-NEXT-STEP

Only one point to note in using loops is that if you use STEP then the values must step exactly to the limit. Thus:

10 FOR A=1 TO 11 STEP 2

will work fine, but:

10 FOR A= 1 TO 11 STEP 3

will never end - an infinite loop occurs.

If you are in some doubt as to whether it will step exactly to the limit, then it is probably adviseable to take the extra time to run the relavent line on its own, and test the final value acheived in Spectrum BASIC to see if it matches the limit. If not, adjust accordingly. Also, use REM B statements so that you can get out of endless loops (REM B tests to see if the BREAK key is pressed).

PAUSE

The values of PAUSE are set up to be much the same as in BASIC. However, you cannot press a key to end PAUSE and thus the use of PAUSE 0 to wait for a key press is <u>not</u> recommended.

READ DATA RESTORE

You can only use single numeric variables after READ, and integers in DATA statements. RESTORE sets the pointer to the start of all DATA - and hence you cannot RESTORE to a specific line number. DATA may not contain, obviously, arrays or strings.

RND

This gives a number from 0 to 32767. You would thus divide RND to get a lower range of numbers. For instance RND/128 will give a range from 0 to 255. For maximum randomness use RANDOMISE RND before using RND itself. Note that RND uses SUPER C's own special psuedo-random number table which is good but not terribly extensive. If you want to approximate 'true' randomness you will have to devise a more elaborate routine with a considerably larger data table. Also note that as in BASIC's RND, the compiler's RND uses the system's variable FRAMES which is updated every 50th of a second. In BASIC this is quite frequent, but for the compiler this is slow. Thus if you poke the attribute file with random numbers you will see that the colours change about every two and a half lines: This is because this is how much SUPER C can do in a 50th of a second!

ARITHMETICAL FUNCTIONS * / - +

These are as per usual except that all equations are calculated

strictly left-to-right. That is, all brackets are ignored. This will cause unexpected results in a few cases if you are not attentive. For instance one might be used to an expression like this:

```
10 POKE 54000, PEEK (9000+34)
```

Here you might expect the location 9034 to be PEEKed, but in fact location 9000 will be and its value will be added to 34.

USR CALLS

USR must be followed by a number or a numeric variable. Else its as usual.

NUMBERS IN GENERAL

Printed numbers can be from 0 to 32767 or -32767 to 0
Inputed numbers can be from 0 to 65536 or -32767 to + 32767

Naturally they are all integers. The enhancement will offer floating point numbers which it is anticipated to be represented by lower case letters for such variables.

LET/ NUMERIC VARIABLES

You can only have integer numeric variables . There can be 26 in all labelled A to Z (capitals only). Note that they are all intially set to zero so unlike BASIC you do not need to define them near the start of a program. This is much more in line with Microsoft BASIC then with Sinclair.

CONDITIONAL STATEMENTS

SUPER C will recognise the usual <>, >, and < but it interprets '<' as 'less than or equal to'.

IF THEN statements must end with a GOTO or GOSUB - not PRINT or POKE or etc. Hence if the result of a condition is to PRINT something, simply GOSUB an address where that PRINT statement can be executed. Similarly for other results of such testing, although LET after THEN is usually all right too.

CLS

This will clear all of the screen - all 24 lines of the display files, not just the upper-most 22 which is cleared by BASIC's CLS command.

CLEAR

This cleare all variables as in BASIC.

RANDOMISE

This must be followed by a number , a numeric variable, or RND.

GOTO/GOSUB/RETURN.

The line jumped to must exist! And GOTO and GOSUB may only be followed by a number - not a calculated line number, or a variable.

Return works as per BASIC to return from a GOSUB, but if it is met independent of a GOSUB it will be interpreted as 'return to BASIC'.

OVER/INVERSE/ETC

All these can only be used on their own and not in PRINT statements. Thus to PRINT in yellow INK on red PAPER you might enter:

```
10 INK6: PAPER 2:PRINT "HI"
```

But remember to change the values of INK and PAPER back again to the general values. This is true for FLASH, BRIGHT,OVER, and INVERSE too.

SOME GENERAL POINTS

1. MULTISTATEMENT LINES.

Clearly these are allowed and are as per BASIC.

2. REMs

You may have noticed that the compiler uses REM statements to enhance BASIC and thus using them is restricted. They may be used to test for the break key being pressed (REM B) or to add machine code routines (REM M,x,y,z,...), or finally to execute a total system 'NEW' (REM N).

3. SCRATCH PAD MEMORY AREA

You are recommended to use the area of memory from 54000 upwards as a scratch pad area, or for storing data, ASCII codes (for strings), and for USR defined characters (see below). This area is not used by the unenhanced compiler.

4. SAVING YOUR COMPILED PROGRAM

To do this you must first return to BASIC. Your program is SAVEd as a machine code one by entering this as a direct command:

SAVE"name" CODE 23500,9268

Don't forget to SAVE any data etc stored above 54000 as well though! Of course you may SAVE the entire 48K memory, but remember that the compiler need not be in memory when the compiled program is being RUN .

Of course, making the compiled program auto-run may be done by setting the first line of BASIC to:

1 SAVE"name" CODE 23500,9268:GOTO4

Assuming that line 4 still contains the RANDOMISE USR CALL. Else, have a GOTO statement to the relavent line number with the CALL in it. You may of course change it from 4 to any other line.

SUPER SPEED

We think that you will probably be amazed at how fast SUPER C compiled programs are. Up to 250 times faster! At least 10 times faster (except when drawing or making a circle as these are the only routines which entail ROM calls).

We made some fairly standard 'Benchmark' tests similar to those published in Personal Computer World when they review a new machine. For the sake of comparison we include the BBC Model B, and the Spectrum BASIC before and after compilation.

		Spectrum	
Benchmark	BBC	BASIC	COMPILED
1	1.0s	4.8	0.15
2	3.1	8.7	0.09
3	8.2	21.1	1.10
4	8.7	20.4	0.99
5	9.1	24.0	1.11
6*	13.9	55.3	1.91
7*	21.4	80.7	2.14

* The last two benchmarks were measured with the compiler by assigning an areaof memory to store values of an array in BM6 and actually POKEd the values there in BM7.

In these cases the range of speed increase is 20 to 100 times, and these are not necessarily the best routines to show off the compiler. The compiler will frequently be as fast as compiled languages such as PASCAL and FORTH - but with the immense advantage that there's no new language to learn. Because we have optimised certain routines (the PLOT and PRINT ones for e.g.) there is every possibility that SUPER C compiled BASIC may be faster than any PASCAL or FORTH program using these commands/statements predominently.

For maximum speed use a variable which is incremented and tested for a limit rather than FOR-NEXT loops. If you have written machine code before , then structuring your BASIC program to be compiled as far as possible in a similar manner to the way you would write machine code will usually produce a faster program.

MOVING BETWEEN BASIC AND COMPILED .CODE

There is generally no problem in doing this, and line 4 tells you what the USR call is to re-enter the compiled program. However, you do only have a limited amount of memory left for BASIC (some 256 bytes or so).

Earlier I mentioned methods of passsing variable values from the BASIC to the COMPILED code. You can always return to BASIC to PRINT to the printer, SAVE,LOAD, set up an array or strings, or do a floating point calculation.

One problem you will encounter is that as the compiled program is entered the screen is cleared. Obviously there will be times when you do not wish this to be so (when you've set up a visual display which used floating point (SIN and COS ,say) to create and want to jump to the code to change the colours - see later. You can alter a couple of memory lo tions using POKE to stop the compiler clearing the screen. POKE these two locations as follows:

```
POKE 25092,184      <ENTER>
POKE 25094,201      <ENTER>
```

This will cause that particular compiled program not to clear the screen when it is RUN, but to make all programs compiled fail to clear the screen you will need to do these POKEs instead:

```
POKE 41476,184
POKE 41478,201
```

For the initiated, this alters the compiler's 'run-time' routines which are incorporated into any compiled program.

SCANNING THE KEYBOARD.

You can use CODE INKEY$ or CHR$ CODE INKEY$ to do this. However, SUPER C will only detect the key pressed at any given moment and thus will not handle shifted keys as one is accustomed. To detect a shifted key you will need to first detect that either SYMBOL SHIFT or CAPS SHIFT has been pressed. The compiler's writer has made this job easy for you, because although they do not usually have codes he has given them the codes 39 (CAPS SHIFT) and 24 (SYMBOL SHIFT). You need only then GOSUB to a routine which checks what is pressed next.

By this method, of course, you may only detect one keypress at a time. By using the IN facility though there is no reason why you should not be able to detect any number of keys pressed in unison. See page 160 of the Spectrum manual.

NUMERIC ARRAYS

Whilst SUPER C cannot handle numeric arrays in the usual way you have been used to in BASIC, you can nonetheless create numeric arrays with a little thought. Naturally they will have to be integer arrays. The method suggested is to set aside a known and restricted block of memory in the free area from 54000 upwards to be the array(s). You then create,in effect, a variable area there by POKEing the values of the array into successive memory locations. Thus an array of ten elements might be designated as occupying locations 54000 to 54009 in memory. Thus the first element will be in 54000 the second in 54001, and so on. This being equivalent to A(1), A(2) and etc. You could even create a variable file in which the name of the array is stored in the first memory location (use CODE 65 for 'A' for e.g.) and then the next location could hold the number of ele ments in that array. Thus DIM A(10) would be the same as POKEing 54000 with 65 (='A') and 54001 with 10 for the length. Array manipulation is thus no more than routines to shift around the contents of memory usingPEEKS and POKES, and recalling items is simply a matter of keeping track of where the items are in memory.

STRING ARRAYS

These can be created in a similar way to the numeric arrays, but using ASCII codes for the characters in the array or string. Thus you might translate the BASIC LET A$="HI THERE" as location 54000 POKEd with 65 (for 'A'), location POKEd with 8 (its length including the space) and then the next ten bytes/memoy locations being the ASCII CODES for 'H', 'I',and so on. See the rear of your Spectrum manual for the ASCII CODES or the rear of Tim Langdell's book 'The Spectrum Handbook'.

Here's a simple example to give you the idea:

BASIC	COMPILER
10 LET A$="HI"	10 POKE 54000,72: POKE54001,73
20 PRINT A$	20 PRINT CHR$ PEEK 54000: PRINT CHR$ PEEK 54001
30 PRINT A$(2)	30 PRINT CHR$ PEEK 54001

As you can see until you can purchase the enhancement to SUPER C arrays and strings will require extra thought. However they

are quite possible, and even rather easy when you become accustomed to thinking in terms of memory locations.

USR DEFINED GRAPHICS CHARACTERS

As you will see if you try to get the compiler to print out the characters with CODEs 0 to 255 it will only print the ASCII characters, rendering the rest as question marks. Note that it doesn't hang up on control codes as BASIC does though. To obtain the block graphics you can quite easily create them as USR defined ones (they are built up by the ROM usually).

The heart of creating and using USR defined graphics with the compiler is in understanding how to use the pointer in systems variables called CHARS. This is an address held in two bytes/ memory locations - 23606, and 23607. If you PEEK theseupon switching on you will see that they are 0 and 60 respectively. Converting these to an address in the usual manner:

```
PRINT 0 + 256*60  = 15360
```

This reveals that CHARS points to a location in the ROM. This is in fact 256 bytes lower than the start of the character table, which itself begins with character 32 (a space). Thus, if you POKE 23606 and 23607 with the relevent numbers to allow CHARS to point to , say 54000 instead of 15360 you can place an entire new character set in this area of high memory such that pressing SPACE would print the first character held in the first 8 bytes from 54000 to 54007 (see Tim Langdell's book on the Spectrum for a full explanation of how to create characters). Locations 54008 to 54015 would hold the bytes for the next character (replacing the 'I'), and so on. Of course, you don't need to stay with this new set - you simply POKE 23606 and 23607 back to their original values again to return to the ROM's character set. Thus you can by changing CHARS at the appropriate moment have almost as many new characters as you have memory to put them in!

EXAMPLE ROUTINES.

SCREEN DUMP/SAVE

You can save an entire screen in high memory , clear the screen and then down-load what you created very easily and quickly using SUPER C.

```
10 CIRCLE 128,88,50
20 LET A=16384
30 LET X=54000
40 LET K= PEEKA
50 POKE X,K
60 IF A>22528 THEN GOTO 80
70 LET A=A+1: LET X=X+1: GOTO 40
80 CLS:PAUSE 100
90 LET A= 16384: LET X=54000
100 LET K= PEEK X
110 POKE A,K
120 LET A=A+1: LET X=X+1
130 IF A> 22527 THEN STOP
140 GOTO 100
```

Of course you could call the screen this way whenever you like. You could have also st ored the attributes file too (let the limit of 'A' go to 23296 instead) to re-create the same graphics and colours.

BORDER FLASH / WHITE NOISE

Compile this and you'll get a flashing boarder with sound effects:

```
10 OUT 254,RND
20 REM B
30 GOTO 10
```

Or this, which gives steadier multicoloured lines up the boarder:

```
10 OUT 254,RND/4096
20 REM B
30 GOTO 10
```

CLS TO A NEW INK AND PAPER COLOUR

This routine when compiled changes the INK and PAPER colours to those defined by 'x' in line 20 . This routine could also change FLASH and BRIGHT too. (See p.106ff of Tim Langdell's Spectrum Book).

```
10 LET A=22528
20 POKE A,x            (where x is the attribute)
30 IF A>23295 THEN STOP
40 LET A=A+1: GOTO 20
```

SUPER-ETCH-A-SKETCH

Here is an etch-a-sketch program such as you've never experienced in BASIC . Compile this:

```
10 LET X=128: LET Y=88
20 OVER 1:PLOT X,Y
30 IF CODE INKEY$ =54 THEN LET Y=Y-1
40 IF CODE INKEY$ =55 THEN LET Y=Y+1
50 IF CODE INKEY$ =53 THEN LET X=X-1
60 IF CODE INKEY$ =56 THEN LET X=X+1
70 REM B
80 GOTO 20
```

You will probably find it necessary to put in a few line to test for the edges of the screen too or else you soon over-run them!

SIMULATING 'INPUT'S

SUPER C does not handle either numeric or string inputs, and hence if you do not have the enhancement package you will need to simulated these. This is not too difficult when dealing with characters, but entering numbers entails rather more thought. SUPER C has the advantage over BASIC that it is equally easy to INPUT anywhere on the screen - not just the bottom lines.

String Inputs

For these you will have to set-up a keyboard scanning routine which puts the inputed ASCII codes into memory locations, and chec .s for ASCII CODE 13 which is the RETURN/ENTER CODE (meaning that the entry is complete).

```
10 LET X=54000
20 POKE X, CODE INKEY$
30 IF CODE INKEY$ =13 THEN GOTO 1000
40 PAUSE 10
50 LET X=X+1
60 GOTO 20


1000 PRINT "START OF MAIN PROGRAM"
1010 PRINT "TO PRINT ENTERED STRING
     TO THE SCREEN"
1020 LET X = 54000
1030 PRINT CHR$ PEEK X
1040 LET X=X+1: IF PEEK X =13 THEN STOP
1050 GOTO 1030
```

You may have to invent variations upon this to suit your needs, but this should at least provide the general idea.

Numeric Inputs

These are rather more difficult. Essentially you will need to define an area of memory to be a 'buffer' to store the inputed CODE numbers prior to deciding what number has been inputed. Say you choose the area from 54500 onwards. Use a similar routine to the above to input the ASCII codes into the buffer area, and once again check for CODE 13 as an end of input point. You may want to check that the inputed CODES are in fact between 48 and 57 (the numbers) and reject other inputs. CODE 48 is zero and 57 is 9, thus the CODE minus 48 gives you the number in question. If you include a counter in your input routine then you will know how many digits are involved in the integer number entered. This is important because you will need to multiply each number by the correct power of ten to turn it into the full integer inputed. Thus if the characters '5' and '7' were the only two inputed, then you know you must multiply the first by 10 and add it to the second. If three numbers were inputed then you will need to multiply the first by 100, the second by 10 and add all three together to obtain the number. Although this may sound complicated, and it certainly is compared to BASIC and the SUPER C enhancement, it is not at all difficult once the concept has been grasped.

HOW MUCH MEMORY CAN I USE?

The compiler effectively occupies the region from around 32767 to 54000, although for technical reasons not given here, it is not always using all this space. Effectively then without the enhancement you will have about 8½ K of RAM for BASIC programming, plus a further 11K or so for data, strings, and graphics in high memory. However, do not forget that the compiler itself need not be in memory when the program is run after compilation. Thus the entire RAM from 32767 to 54000 can also be used for data, etc as soon as the main program has been compiled. Obviously testing your program will entail destroying SUPER C, but you will then be able to have a full 48K machine code program.

It is anticipated at the time of writing that the enhancement will enable you to have anything up to twice as much BASIC program to compile.

IS THE CODE PRODUCED RELOCATABLE?

The simple answer is no, not really. The calls made within compiled programs are to specific memory locations and thus the code cannot be simply loaded back into a different area of memory. Hence compiling several small routines to be eventually run side-by-side will not be possible. If you are very adept at machine code you might be able to go through the entire compiled code and change all the jumps - but be warned this would be along task.

WHAT NEXT? - THE FUTURE

You will have gathered that an enhancement is available for SUPER C which enables it to handle strings, arrays, inputs, and floating point arithmetic and functions. This is anticipated to be available in March 1983. Please contact us by letter for further details. It will almost undoubtably contain many other enhancements too which will further extend Sinclair BASIC capabilities.

NOW however, you can obtain at least four further Softek products which will of be of great use to you when using the compiler.

1. TOOLKIT This is a quite exceptional toolkit and is part of our SOFSYS$_{TM}$ SYSTEM along with SUPER C . It has facility to remov-

REM statements (which are reserved for special pruposes by SUPER C , and which add unecessary bytes to BASIC programs nearing RAMTOP limits), it can do a total renumbering of a program from a given starting line number in a given step size. It can also delete specified line numbers, search for any character, number, keyword (and it conveniently lists the lines in which they occur), it can replace almost anything with anything else of the same number of bytes, it can display the current variabels in use and their values, it can give the program size, or variables file size, or the free memory. On top of all this it offers a unique TRACE facility which displays the current line number and statement being executed as the program runs. You can even set the speed at which the program (and hence trace) runs.. Only £9.95 at time of writing.

GRAPHICS KIT This is the ultimate graphics kit for the ZX Spectrum, and can be located in memory along with SUPER C, as can the toolkit. This kit offeres scrolling by pixels or bytes in any direction (up, down, left, right and the diagonals), it will instnadly change the attributes without affecting the display, it will invert the screen, and allows you to store additional screens (or thirds of them) in memory for instant down-loading the the display file tocreat cartoon effects etc. Along with these and other facilities, it offers SPRITE graphics, enabling you to define large charcters in memory which it can move around the screen fast and extremely smoothly. It can even allow you to define windows to be scrolled independently! Only £9.95 at the time of writing.

SOFMON This is a rather excellent disassembler/monitor used for developing many Softek programs. It has around 25 functions and facilities. For instance, it has a full (an errorless!) disassembly of memory into standard Z80A mnemonics. It can show areas in HEX, display them as ASCII codes, find bytes in memory, make space in m/c routines, or close gaps, and it can convert HEX to DEC or vice versa. It has a superb Modify in HEX routine which actually disassembles your inputs as you enter them! It even allows you to type in ASCII text too. It is very user friendly and makes use of Sinclair BASIC-like comands and prompts (such as Scroll?, and the 'K' cursor, as well as pritning on the screen full words when any key is pressed. Only £9.95 at the time of writing.

SOFTIME - THE MASTER KEY PROGRAM

This is the heart of SOFSYS and is a quite astounding program. Softek again acheives the improbable by offering you a real-time clock which is on display even when writing or running another program, as well as the option to have USR DEFINABLE KEYS! Yes, you can define keys 1 to 7 to have up to 100 bytes of BASIC executed immediately you press them whilst holding down the SPACE key. It also offers ON ERROR (key 8) and ON BREAK (key 9). It comes with keys pre-defined so that at a single press you may enter SOFMON, or Compile a program, or enter a toolkit routine, or toggle the SOFTIME clock on or off the screen. Softime even

OTHER SOFTEK SOFTWARE

We have a superb ASSEMBLER coming to complete Sofsys - write for details, for this will be totally in machine code, allow entry of op codes in line numbers and will feature a renumber feature of these lines.

We have a full range of games such as our lauded Meteoroids - hailed the best at only £4.95. And the only truely playable 16K adventure for the ZX Spectrum,called The Zolan Adventure - only £4.95. We have an amazing 48K 3D 'Mazenture' called the 'Dragon's Lair' for only £9.95. And a host of arcade games on the way (such as missile command, centipede, and not just a few surprises!)

SOFTEK SOFTWARE
329 CROXTED ROAD LONDON SE 24, ENGLAND.

This tape stores a BACKGAMMON program suitable for the 48K version of the ZX Spectrum. To LOAD it follow the instructions in the Sinclair manual. Switch on the computer, press the J key to give 'LOAD' on the screen, then the P key twice, whilst holding down SYMBOL SHIFT, to give "", then ENTER: start playing the tape. Stop the tape only when the message 'Wait until machine code loaded' disappears.

To end a game and start another - type QUIT.

Playing instructions are given on the reverse of this inlay card and at the start of the program.

The rules of the International Backgammon Association are summarized below.

1. For the opening throw each player throws a single die. Every tie requires another opening throw. Whoever throws the higher number wins and for his first move plays the numbers upon both dice. After that each player in turn throws two dice.

2. The play of men consists of:

   a) Moving a man (or men) the exact number of points indicated by the number of the two dice thrown. One man may be moved the total of the two dice thrown, or one man may be moved the number shown on one die, and an entirely different man the number on the other die.

   b) Entering a man, in the adversary's inner table, on a point corresponding to the number on a die.

   c) Bearing off a man in the player's inner table, when no man is left outside that table or on the bar, in accordance with rule 8.

[illegible] may be played, but the second number must be played.

5. No play may be made which lands or touches down on a point held by two or more of the adversary's men.

6. When play lands on a point occupied by a single man (a blot) of the adversary such a man is 'hit' and is placed on the bar to await entry according to rule 2b.

7. A player having a man on the bar may not play any other man until that man has been entered.

8. When in a position to bear off you may bear off a man from a point corresponding to the number on a die thrown or from the highest occupied point if that is lower than the number on the die. If a number is thrown for an unoccupied point no man below can be borne off, using such a number, while any man remains on a higher point. You are not required to bear off if you can move a man forward on the board. Here, as always, rule 4 applies.

--o0o--

BACKGAMMON FOR BEGINNERS

In this game you play the WHITE men and your opponent, the computer, plays the BLACK. The game starts with the men set on the predetermined points around the board, each point is labelled with a letter. The purpose of the game is for you to move your men anti-clockwise around the board until they are all in your "inner table" (points f to a). Once all your men are in your inner table your may then 'bear off' i.e. move your men off the board. Your opponent will be moving his men clockwise round the board to his inner table (points s to x) and will then bear off. You move your men one at a time and may move them forward by the numbers shown on the dice. You may move onto any point that does not contain 2 or more of your opponent's men. If you move onto a point which is occupied by only one of your opponents men, your opponent's man will be placed on the bar (the green bar down the centre of the board) and that man must start again from the beginning. You must therefore be aware that, if you leave a single man on a point, this may give your opponent the opportunity to land on that point and send your man to the bar. The first player to bear off all his men is the winner.

TO PLAY

To start the game the dice are rolled and whoever "throws" the highest number starts - using the numbers shown on the dice (Rule 1). Your opponent's is the top die and yours is the bottom die, therefore if the top die has the highest number your opponent starts and if the bottom die has the highest number then you start. If it is a double press ENTER to throw again. When it is your turn - the prompt "Your move?" will be displayed on the screen. You may then move your men by the number shown on the dice. To move a man from one point to another enter the point letter and a number shown on one of the dice, e.g. to move a man forward 4 points from point K enter 'K4' or "4K". If you enter an illegal move you will be told to press ENTER, to recover and must then enter a legal move.

You must move your men by the numbers shown on the dice. e.g. if you throw 4 and 2, you must move one man forward 4 points and another man 2 points, or you may move one man forward 4 and then 2 points (or 2 and then 4).

If you throw a double you may then move double the total shown on the dice. e.g. double 3 allows you to make four moves of 3. (see Rules 2-5)

THE BAR

While you have any men on the bar you may not move any of your men which are on the board. (Rule 7) To move off the bar onto the board you must enter the bar letter 'z' followed by one of the numbers shown on the dice. This must be to a point which is either empty, occupied by your own men or one of your opponent's men - in which case you may send that man to the bar. If you are unable to move from the bar because the numbers on the dice would land you on a point which is occupied by two or more of your opponent's men - then you forfeit your turn.

BEARING OFF

Once all your men are on your inner table you may then bear off (see Rule 8). However, if for any reason one of your men is placed on the bar by your opponent after all your men have reached the inner table you may not bear off any more men until that man has returned to your inner table.

--o0o--

This program uses machine code evaluation for fast response. High resolution colour display with on-screen dice roll. Includes instructions to enable beginners to

SPECTRUM BACKGAMMON

This tape stores a BACKGAMMON program suitable for the 48k version of the ZX Spectrum. To LOAD it follow the instructions in the Sinclair manual. Switch on the computer, press the J key to give 'LOAD' on the screen, then the P key twice, whilst holding down SYMBOL SHIFT, to give "", then ENTER: start playing the tape. Stop the tape only when the message 'Wait until machine code loaded' disappears.

To end a game and start another - type QUIT.

Playing instructions are given on the reverse of this inlay card and at the start of the program.

The rules of the International Backgammon Association are summarized below.

1. For the opening throw each player throws a single die. Every tie requires another opening throw. Whoever throws the higher number wins and for his first move plays the numbers upon both dice. After that each player in turn throws two dice.

2. The play of men consists of:

   a) Moving a man (or men) the exact number of points indicated by the number of the two dice thrown. One man may be moved the total of the two dice thrown, or one man may be moved the number shown on one die, and an entirely different man the number on the other die.

   b) Entering a man, in the adversary's inner table, on a point corresponding to the number on a die.

   c) Bearing off a man in the player's inner table, when no man is left outside that table or on the bar, in accordance with rule 8.

3. Doubles require four plays of the number on the dice.

4. Plays must be made for both dice if possible. Either number may be played first. If either number may be played, but not both, then the higher number must be played.

5. No play may be made which lands or touches down on a point held by two or more of the adversary's men.

6. When play lands on a point occupied by a single man (a blot) of the adversary such a man is 'hit' and is placed on the bar to await entry according to rule 2b.

7. A player having a man on the bar may not play any other man until that man has been entered.

8. When in a position to bear off you may bear off a man from a point corresponding to the number on a die thrown or from the highest occupied point if that is lower than the number on the die. If a number is thrown for an unoccupied point no man below can be borne off, using such a number, while any man remains on a higher point. You are not required to bear off if you can move a man forward on the board. Here, as always, rule 4 applies.

--oOo--

CP SOFTWARE, 17 Orchard Lane, Prestwood, Great Missenden, BUCKS, HP16 9NN

BACKGAMMON FOR BEGINNERS

In this game you play the WHITE men and your opponent, the computer, plays the BLACK. The game starts with the men set on the predetermined points around the board, each point is labelled with a letter. The purpose of the game is for you to move your men anti-clockwise around the board until they are all in your "inner table" (points f to a). Once all your men are in your inner table you may then "bear off" i.e. move your men off the board. Your opponent will be moving his men clockwise round the board to his inner table (points s to x) and will then bear off. You move your men one at a time and may move them forward by the numbers shown on the dice. You may move onto any point that does not contain 2 or more of your opponent's men. If you move onto a point which is occupied by only one of your opponents men, your opponent's man will be placed on the bar (the green bar down the centre of the board) and that man must start again from the beginning. You must therefore be aware that, if you leave a single man on a point, this may give your opponent the opportunity to land on that point and send your men to the bar. The first player to bear off all his men is the winner.

TO PLAY

To start the game the dice are rolled and whoever "throws" the highest number starts - using the numbers shown on the dice (Rule 1). Your opponent's is the top die and yours is the bottom die, therefore if the top die has the highest number your opponent starts and if the bottom die has the highest number then you start. If it is a double press ENTER to throw again. When it is your turn - the prompt "Your move?" will be displayed on the screen. You may then move your men by the number shown on the dice. To move a man from one point to another enter the point letter and a number shown on one of the dice e.g. to move a man forward 4 points from point K enter "K4" or "4k". If you enter an illegal move you will be told to press ENTER, to recover and must then enter a legal move.

You must move your men by the numbers shown on the dice. e.g. if you throw 4 and 2, you must move one man forward 4 points and another man 2 points, or you may move one man forward 4 and then 2 points (or 2 and then 4).

If you throw a double you may then move double the total shown on the dice. e.g. double 3 allows you to make four moves of 3. (see Rules 2-5)

THE BAR

While you have any men on the bar you may not move any of your men which are on the board. (Rule 7) To move off the bar onto the board you must enter the bar letter 'z' followed by one of the numbers shown on the dice. This must be to a point which is either empty, occupied by your own men or one of your opponent's men - in which case you may send that man to the bar. If you are unable to move from the bar because the numbers on the dice would land you on a point which is occupied by two or more of your opponent's men - then you forfeit your turn.

BEARING OFF

Once all your men are on your inner table you may then bear off (see Rule 8). However, if for any reason one of your men is placed on the bar by your opponent after all your men have reached the inner table you may not bear off any more men until that man has returned to your inner table.

--oOo--

This tape stores a BACKGAMMON program suitable for the 48K version of the ZX Spectrum. To LOAD it follow the instructions in the Sinclair manual. Switch on the computer press the J key to give 'LOAD' on the computer screen then the P key twice, whilst holding down SYMBOL SHIFT to give "". Then ENTER: start playing when the tape has been only when the message 'Wait until machine code loaded' disappears.

To end a game and start another - type QUIT.

Playing instructions are given on the reverse of this inlay card and at the start of the program.

The rules of the International Backgammon Association are summarised below.

1. For the opening throw each player throws a single die. Every tie requires another opening throw. Whoever throws the higher number wins and for his first move plays the numbers upon both dice. After that each player in turn throws two dice.

2. The play of men consists of:

   a) Moving a man (or men) the exact number of points indicated by the number of the two dice thrown. One man may be moved the total of the two dice thrown, or one man may be moved the number shown on one die, and an entirely different man the number on the other die.

   b) Entering a man, in the adversary's inner table, on a point corresponding to the number on a die.

   c) Bearing off a man in the player's inner table, when no man is left outside that table or on the bar, in accordance with rule 8.

3. No play may be made which lands or touches down on a point held by two or more of the adversary's men.

4. When play lands on a point occupied by a single man (a blot) of the adversary such a man is 'hit' and is placed on the bar to await entry according to rule 2b.

5. A player having a man on the bar may not play any other man until that man has been entered.

6. When in a position to bear off you may bear off a man from a point corresponding to the number on a die thrown or from the highest occupied point if that is lower than the number on the die. If a number is thrown for an unoccupied point no man below can be borne off, using such a number, while any man remains on a higher point. You are not required to bear off if you can move a man forward on the board. Here, as always, rule 4 applies.

--oOo--

The material on this tape and documentation, or any part thereof, shall not be copied for use by any other person or organisation, neither shall it be loaned nor hired.

BACKGAMMON FOR BEGINNERS

In this game you play the WHITE men and your opponent, the computer, plays the BLACK. The game starts with the men set on the predetermined points around the board, each point is labelled with a letter. The purpose of the game is for you to move your men anti-clockwise around the board until they are all in your "inner table" (points f to a). Once all your men are in your inner table your may then "bear off" i.e. move your men off the board. Your opponent will be moving his men clockwise round the board to his inner table (points x to z) and will then bear off. You move your men one at a time and may move them forward by the numbers shown on the dice. You may move onto any point that does not contain 2 or more of your opponent's men. If you move onto a point which is occupied by only one of your opponents men, your opponent's man will be placed on the bar (the green bar down the centre of the board) and that man must start again from the beginning. You must therefore be aware that, if you leave a single man on a point, this may give your opponent the opportunity to land on that point and send your man to the bar. The first player to bear off all his men is the winner.

TO PLAY

To start the game the dice are rolled and whoever "throws" the highest number starts - using the numbers shown on the dice (Rule 1). Your opponent's is the top die and yours is the bottom die, therefore if the top die has the highest number your opponent starts and if the bottom die has the highest number then you start. If it is a double press ENTER to throw again. When it is your turn - the prompt "Your move?" will be displayed on the screen. You may then move your men by the number shown on the dice. To move a man from one point to another enter the point letter and a number shown on one of the dice. e.g. to move a man forward 4 points from point K enter "K4" or "4K". If you enter an illegal move you will be told to press ENTER, to recover and must then enter a legal move.

You must move your men by the numbers shown on the dice. e.g. if you throw 4 and 2, you must move one man forward 4 points and another man 2 points, or you may move one man forward 4 and then 2 points (or 2 and then 4).

If you throw a double you may then move double the total shown on the dice. e.g. double 3 allows you to make four moves of 3. (see Rules 2-6)

THE BAR

While you have any men on the bar you may not move any of your men which are on the board. (Rule 7) To move off the bar onto the board you must enter the bar letter 'z' followed by one of the numbers shown on the dice. This must be to a point which is either empty, occupied by your own men or one of your opponent's men - in which case you may send that man to the bar. If you are unable to move from the bar because the numbers on the dice would land you on a point which is occupied by two or more of your opponent's men - then you forfeit your turn.

BEARING OFF

Once all your men are on your inner table you may then bear off (see Rule 8). However, if for any reason one of your men is placed on the bar by your opponent after all your men have reached the inner table you may not bear off any more men until that man has returned to your inner table.

--oOo--

n° 11

TO LOAD A PROGRAM

The procedure for loading a program is described in chapter 6 of the Sinclair SPECTRUM Instruction Booklet. For your convenience the procedure is summarized here.

1. Position the tape before the start of the program.
2. Ensure that the EAR socket on the SPECTRUM is connected to the earphone socket on your tape recorder.
3. Set the volume control to about 3/4 of maximum and any tone controls to: treble high; bass low. Alternatively, use levels that you have already found to be successful on your recorder.
4. Enter LOAD "" CODE, by pressing the following keys - J (for LOAD) symbol shift and P twice (for "") and caps shift with symbol shift followed by I (for CODE) and press the ENTER key.
5. Start the tape recorder playing.

If the program fails to load, approximate load time, 2 3/4 minutes, press BREAK and try again with a new volume setting. If the program still fails to load refer to chapter 6 of the Sinclair SPECTRUM Instruction Booklet.

CP SOFTWARE, 17 Orchard Lane, Prestwood, Great Missenden, BUCKS, HP16 9NN

ABOUT 'SUPERCHESS II'

'SUPERCHESS II' has 7 levels of play numbered 2 to 8. Starting at level 2 the play becomes progressively stronger with each successive level.

Overall response times for the first 5 levels (2 to 6) are within tournament time limits. For example, a typical response time at level 2 is 1 seconds, and at level 3 is 8 seconds. The actual response time to any one move depends not only on the level but also on the complexity of the position, just as when playing another chess player. The higher levels of play are suitable for extended games, similar to postal chess, and for problem solving.

In the same way that many players learn several standard openings, 'SUPERCHESS II' is programmed with several standard openings.

All legal moves are allowed including castling and pawn capture en passant. If, by mistake or design, you try to enter an illegal move it will not be accepted.

The standard algebraic chess notation is used to enter and display moves. The co-ordinate system for this notation is included as part of the board display to assist those unfamiliar with the notation. The instructions which follow give full details for entering moves and selecting from the available options.

The board display makes use of the SPECTRUM's high resolution graphics and includes a feature where you can select the colours of the display. White always starts at the bottom of the display and, as a reminder, the white player is indicated below the board, for example, if the SPECTRUM is white then the message - 'I am WHITE' is displayed.

The display also includes a list of the previous (up to 10) moves, the playing level.

'SUPERCHESS II' also provides several other features, which can be selected either before or during a game, these are:

HELP, this lists the available options.

RECOMMENDED MOVE, if you can't decide on your next move then, this option recommends a move that you might play.

SELF PLAY, allows you to sit back and watch 'SUPERCHESS II' play itself.

RESIGN, this is the option to use when you consider the game is lost, you may get quite familiar with this!

ANALYSE, allows you to set up and/or change a position by adding and removing pieces.

TECHNICAL INFORMATION, gives brief details of the programs evaluation strategy.

COLOUR CHANGE, this option allows you to select the colours of the display.

PLAYING INSTRUCTIONS

To load and run enter LOAD "" CODE - ENTER.

The program is self starting on completion of loading but before a game can start you are asked to select colour, playing level etc. Your responses to these questions should be entered using only the first letter, example 'Do you want Black or White?' if you choose white you should only enter "w"-ENTER. All responses and moves must be completed by pressing ENTER.

Before a game you will be asked:-

'Do you want to "Play" or "Analyse"?'

(here we assume play is selected)

'Do you want "Black" or "White"?'

'Enter playing level 2 to 8'

When you have entered the appropriate responses the board will be displayed and all is ready to start the game. In fact if you are black the first move will have been made.

Moves are made by entering first the coordinates of the square you wish to move FROM followed by the coordinates of the square you wish to move TO. Example, White wishes to move his queen's pawn from square D2 to square D4, enter "d2d4" (don't forget to press ENTER). Note - lowercase letters must be used to enter moves. If you try to enter an illegal move you will get the message 'INVALID MOVE' and you should then enter a correct move.

Captures are made in the same way as moves. Example, to capture a bishop on F6 with a knight on E4, enter "e4f6". To castle only the king move is entered. Example, to castle kingside enter "e1g1", to castle queenside enter "e1c1". Pawn capture en passant is entered as for other pawn captures. Example, to use a pawn on C5 to capture a pawn that has just moved from B7 to B5 enter "c5b6". On reaching the 'back rank' pawns are automatically promoted to a queen.

Once a game has ended either with a checkmate or if you choose to resign you will be asked if you want another game. Entering "y" for YES returns to the start options while entering "no" returns to basic and clears the program.

During Play, several further options are available.

'Recommended move', which can be used only after the opening. Enter "rec" and a move will be suggested. You may now enter this or any other (legal) move you choose.

You can change:- 'playing level' by entering "lev" followed by the number of the level required; the colours of the display by entering "col", you are then asked to select the colour of the pieces and squares.

'Resign', when you feel the game is lost or if you just want to stop the game enter "res", you will then be asked if you want another game.

'Help', used once the game has started or in the Analysis mode, enter "hel" to obtain a list of options. to get back to the game enter "p".

'Self Play', can be used only after the opening, to obtain enter "sel", now 'SUPERCHESS II' will play itself but you must press ENTER after each move. To get back to 'Play' enter "sel".

Technical information, gives brief details of the program's evaluation strategy, enter "tec".

ANALYSE

You can select 'Analyse' before a game starts by entering "a" or during play by entering "ana". The board and existing position will be displayed. The board can be cleared (except for the two kings) by entering "cl". This can be done whenever the message 'enter "cl" to clear board' is displayed.

'Analyse' allows you to set up and change positions as required. To do this, you are first asked to enter the square that you wish to add a piece to or remove a piece from, this is followed by a request to enter the colour and piece code of the piece you want on that square (see table). To remove a piece instead of entering the colour and piece code enter "p". Each type of piece has a numerical code which must be preceded by the colour when entering. The table and prompts of what to enter are included on the display. The complete code is the colour followed by a number:-

Black	White	Pawn	Knight	Bishop
b	w	1	2	3

Rook	Queen	King	Remove
4	5	6	p

Examples:-

To put a white knight to E4 enter "e4" and then enter "w2"

To remove a black bishop from C2 enter "c2", then enter "p"

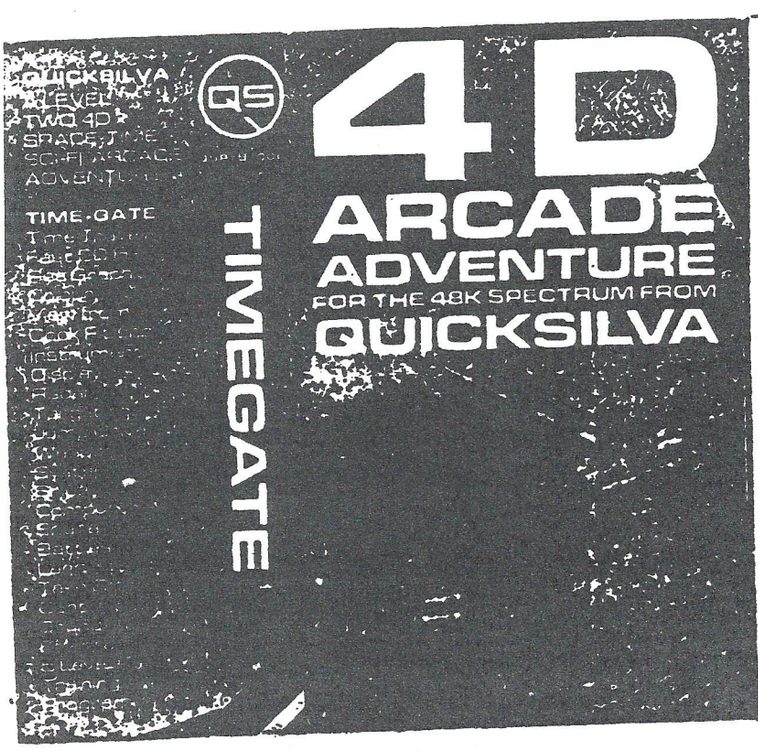
To move a white queen from D1 to D5 enter "d5", then enter "w5", next enter "d1" and finally enter "p"

One use of 'Analyse' is to set up a position and then play from there.

To return to 'Play' enter "a" followed by a move or if you want to start a new game enter "a" followed by "res" followed by "y" and this will return you to the start.

11

B1



LOADING
Time-Gate
You will find the procedure for loading a program in the Spectrum basic manual chapter 20. The procedure for loading Time-Gate is given below:

1. Connect the ear socket of the Spectrum to the ear socket of your cassette recorder.
2. Make sure the tape is wound to the beginning.
3. Set the volume control to a suitable level.
4. Set maximum treble, minimum bass on the tone controls.
5. Type LOAD "" CODE
6. Press ENTER
7. Start the cassette recorder.
8. The program will RUN itself once loaded and will provide instructions.

If the program does not load correctly try a different volume level.

TIMEGATE

Time stretches before and behind me in century long strands measuring the age of the stars, the age of men, but irrelevant to me. I am lost down the years, stuck in a limbo where I am unique: A man without age, an anachronism of the future thrown into the past. My ship is carrying me faster than light through the stars at a speed with no meaning; stars swirl around me, sometimes I feel I could reach out and touch them; other times I feel alone and lost, their light twinkling with malicious laughter. At these speeds lost in the distant past the only other beings I have any contact with are the Enemy. With the Enemy, their ships as fast as mine, I engage in macabre dances of death firing out energies at them until they are with me no more; their debris scattered and receding from me in time and space.

That is my mission, to travel back along the trail of ravaged planets, following the Enemy back through the millenia to where their aggression was given birth and they first reached out to grasp and strangle the peoples of the Universe, for the Enemy must be stopped. Stopped before the first blow was struck, before blood and tears flowed together under the hand of the Enemy.

I search back through time by passing through the TIME-GATES, portals scattered throughout the fabric of time and space which link the future with the past. The TIME-GATES are the key to my mission. The TIME-GATES, enigmas which only lightly brush at the edges of reality, the bulk of their mechanisms lost in some distant dimension. Through these I travel, flung this way and that by the capricious currents of Time. Spat out into a new sector of space, in a different era I can always be sure of the Enemy, they are constant, returning to attempt my destruction again and again. I fight my way through them to planets where I can rest briefly before continuing my mission. Each planet different, an adventure in itself if I only had more of that thing which has so little meaning for me: Time. Each planet unique but almost inveriably scarred by the Enemy. I cannot rest when surrounded by the bitter fruits of the Enemy's conquest, I must force myself on in search of the Brood of the Enemy.

I am confident that I will complete my mission and find peace at some time in the past, but until then I am one man against an empire, one man adrift in an ocean of time punctuated by the enigmatic TIME-GATES.

GROUND ATTACK LOAD INSTRUCTIONS
For comprehensive loading instructions consult P24 of the Spectrum instruction manual.
To load GROUND ATTACK
1. Ensure that the ear socket on the Spectrum is connected to the cassette ear socket.
2. Rewind the tape.
3. Set the volume and tone controls to a suitable level.
4. Type LOAD ""
5. Press ENTER.
6. Start the cassette recorder.
7. Once loaded the game will RUN automatically.
8. Good Luck!!

PLAYING INSTRUCTIONS
Your mission is to pilot your spaceship through tortuous caverns while destroying the enemy missile launchers and fuel dumps. Endurance as well as quick thinking are needed to survive. No-one has yet succeeded. 9-skill levels.
Mission Controls

5—Decelerate,	8—Accelerate,
6—Dive,	9—Drop Bombs,
7—Climb,	0—Fire lasers.

B1/15