

Table of Contents

Type 0 - Player.....	1
Type 1 - Bullet	6
Type 2 - Zombie.....	8
Type 3 - Ammo Box.....	10
Type 4 - Heart.....	11
Type 5 - Virus (Zombie bullet).....	11
Initialise sprite.....	13
Main loop 1	13
Main loop 2	15
Game initialisation	23
Restart screen	24

Type 0 - Player	
IF SCREEN = 1	
EXIT	
ENDIF	
IF J <> 255	If the screen has finished drawing, and
IF J > 199	if the indicator is at least 200, and
IF J = 253	if the indicator is 253, then
LET X 88	relocate Bob's sprite
LET Y 120	
LET IMAGE 0	and reset its image.
LET FRAME 1	
COLOUR 79	BRIGHT WHITE on BLUE
LET LINE 1	
LET COLUMN 24	
MESSAGE 8	Initial score ("00000000").
ENDIF	
ADD 1 TO J	Increment J indicator.
ENDIF	
ELSE	
LET E 0	Reset the item count indicator.
REPEAT 2	
ADD 1 TO E	Increment the item indicator.
GETRANDOM 28	This section gets a random position for
ADD 1 TO RND	any sprite that may be spawned (in this case,
LET N RND	either the ammo box or the heart), and ensures
IF N = 5	it is safe so that the spawned sprite does not
SUBTRACT 1 TO N	appear on top of the maze wall...
ENDIF	
IF N = 6	
ADD 1 TO N	
ENDIF	
IF N = 11	
SUBTRACT 1 TO N	
ENDIF	
IF N = 12	
ADD 1 TO N	
ENDIF	
IF N = 18	
SUBTRACT 1 TO N	
ENDIF	
IF N = 19	
ADD 1 TO N	
ENDIF	
IF N = 24	
SUBTRACT 1 TO N	
ENDIF	
IF N = 25	
ADD 1 TO N	
ENDIF	
GETRANDOM 17	
ADD 3 TO RND	
LET M RND	
IF M = 8	
SUBTRACT 1 TO M	
ENDIF	
IF M = 9	

ADD 1 TO M	
ENDIF	
IF M = 14	
SUBTRACT 1 TO M	
ENDIF	
IF M = 15	
ADD 1 TO M	
ENDIF	...all done.
MULTIPLY M BY 8	Calculate screen X
MULTIPLY N BY 8	and Y co-ordinates.
LET F 0	Reset the sprite spawned indicator.
IF E = 1	If this is the ammo box, then
LET K PARAMA	
ADD PARAMB TO K	If the player has less bullets in their
IF C > K	pistol than the number of sprites on screen,
SPAWN 3 10	then spawn the ammo box (type 3, image 10)
LET F 1	Set the sprite spawned indicator.
ENDIF	
ENDIF	
IF E = 2	If this is the heart, then
IF LIVES <= 2	If the player does not have enough lives, then
SPAWN 4 8	spawn a heart (type 4, image 8).
SPAWNED	
LET FRAME 0	
LET PARAMB 0	
ORIGINAL	
LET F 1	Set the sprite spawned indicator.
ENDIF	
ENDIF	
IF F = 1	If a sprite has been spawned, then
SPAWNED	
LET X M	move it to the already calculated safe random
LET Y N	position.
ORIGINAL	
ENDIF	
ENDREPEAT	
LET K 0	
LET J 0	Reset the screen draw indicator.
LET E C	Store number of sprites (for clearance bonus).
ENDIF	
IF KEY 5	If the PAUSE key ("H") has been pressed,
COLOUR 135	Flashing BLACK and WHITE
LET LINE 0	
LET COLUMN 0	
MESSAGE 7	"PAUSED - Press ANY key"
WAITKEY	
COLOUR 78	WHITE
LET LINE 0	
LET COLUMN 0	
MESSAGE 10	"The Adventures of Bouncing Bob"
ENDIF	
IF KEY 6	If the RELOAD key ("R") has been pressed,
IF GOT 0	and if the Bob has the pistol,
IF PARAMA = 0	and if the pistol is empty,
IF PARAMB > 0	
IF PARAMB > 6	and if the player has extra ammo, then:

LET K 6	Pistol can only hold max 6 bullets.
ELSE	
LET K PARAMB	Pistol can hold all available bullets.
ENDIF	
ADD K TO PARAMA	Add K bullets to the pistol, and
SUBTRACT K FROM PARAMB	subtract K bullets from the ammo.
BEEP 100	
LET RND 26	Temporary column counter.
COLOUR 87	WHITE on RED
REPEAT PARAMA	Draw all available bullets on screen...
LET COLUMN RND	
LET LINE 23	
MESSAGE 12	Bullet.
ADD 1 TO RND	
ENDREPEAT	...done.
LET LINE 23	
LET COLUMN 21	
COLOUR 87	
MESSAGE 13	"Bullet x 00"
IF PARAMB <= 9	
LET COLUMN 24	
ELSE	
LET COLUMN 23	
ENDIF	
LET LINE 23	
DISPLAY PARAMB	Display how much extra ammo is available.
ENDIF	
ENDIF	
ENDIF	
IF J = 0	If not drawing the maze,
IF IMAGE = 7	If Bob has been killed,
IF A = 0	If it is time to animate the sprite,
BEEP 2	
ANIMATE	
ENDIF	
IF FRAME = 12	If the Bob dying animation has completed,
IF LIVES > 0	and if Bob has more lives, then
LET IMAGE 0	reset Bob's sprite image and
LET FRAME 0	frame number.
ELSE	
KILL	Otherwise execute the Kill event.
ENDIF	
ENDIF	
ELSE	Otherwise, if Bob is still alive
DETECTOBJ	
IF OBJ <> 255	
BEEP 60	
GET OBJ	
IF OBJ = 0	If the pistol has been collected,
LET LINE 23	
LET COLUMN 21	
COLOUR 87	WHITE on RED
MESSAGE 13	"Bullet x 00"
LET RND 26	Temporary column counter.
REPEAT 6	Draw all available bullets on screen...

LET LINE 23	
LET COLUMN RND	
MESSAGE 12	Bullet.
ADD 1 TO RND	
ENDREPEAT	...done.
LET PARAMA 6	Update the ammo counter.
ENDIF	
LET LINE 23	
LET COLUMN 21	
COLOUR 87	
MESSAGE 13	"Bullet x 00"
IF PARAMB <= 9	
LET COLUMN 24	
ELSE	
LET COLUMN 23	
ENDIF	
LET LINE 23	
DISPLAY PARAMB	Display how much extra ammo is available.
ENDIF	
LET R 1	Assume sprite will be animated.
IF KEY 0	If the RIGHT key ('P') has been pressed,
IF CANGORIGHT	
SPRITERIGHT	
LET IMAGE 2	Change sprite to Bob moving right.
LET DIRECTION 2	Right!
ENDIF	
IF Y > 238	If Bob is at the right hand edge of the screen,
ADD 1 TO 0	move to the next screen on the right.
IF 0 = 129	If this is the last screen, then
LET 0 1	cycle round to the first screen on the left.
ENDIF	
LET Y 10	Position Bob on the left of the next screen.
LET J 10	Draw the next screen.
ENDIF	
ENDIF	
IF KEY 1	If the LEFT key ('O') has been pressed,
IF CANGOLEFT	
SPRITELEFT	
LET IMAGE 1	Change sprite to Bob moving left.
LET DIRECTION 4	Left!
ENDIF	
IF Y = 0	If Bob is at the left hand edge of the screen,
SUBTRACT 1 TO 0	move to the next screen on the left.
IF 0 = 0	If this is the first screen, then
LET 0 128	cycle round to the last screen on the right.
ENDIF	
LET Y 228	Position Bob on the right of the next screen.
LET J 10	Draw the next screen.
ENDIF	
ENDIF	
IF KEY 3	If the UP key ('Q') has been pressed,
IF CANGOUP	
SPRITEUP	
LET IMAGE 3	Change sprite to Bob moving up.
LET FRAME 0	Show the middle sprite image.
LET R 0	Do not animate this sprite.

LET DIRECTION 1	Up!
ENDIF	
IF X <= 16	If Bob is at the top edge of the screen,
SUBTRACT 1 TO P	move to the next screen above.
IF P = 0	If this is the first screen, then
LET P 127	cycle round to the last screen at the bottom.
ENDIF	
LET X 160	Position Bob on the bottom of the next screen.
LET J 10	Draw the next screen.
ENDIF	
ENDIF	
IF KEY 2	If the DOWN key ('A') has been pressed,
IF CANGODOWN	
SPRITEDOWN	
LET IMAGE 0	Change sprite to Bob moving down.
LET FRAME 2	Show the middle sprite image.
LET R 0	Do not animate this sprite.
LET DIRECTION 3	Down!
ENDIF	
IF X > 166	If Bob is at the bottom edge of the screen,
ADD 1 TO P	move to the next screen below.
IF P = 128	If this is the last screen, then
LET P 1	cycle round to the first screen at the top.
ENDIF	
LET X 26	Position Bob on the top of the next screen.
LET J 10	Draw the next screen.
ENDIF	
ENDIF	
IF KEY 4	If the FIRE key ('SPACE') has been pressed,
IF GOT 0	Does the player have the gun?
IF PARAMA > 0	Is the gun loaded?
IF D = 0	Only fire if not already firing.
IF C <= 11	
IF B = 0	
LET B 5	
ENDIF	
ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF B = 5	If Bob is firing the pistol,
IF DIRECTION = 1	and if Bob is moving up, and
IF CANGOUP	if there is space above Bob, then
SUBTRACT 8 TO X	Reposition for bullet.
SPAWN 1 4	Type 1, Image 4 (Bullet)
SPAWNED	
LET FRAME 0	Set the frame for this bullet (up).
LET DIRECTION 1	Set the direction for this bullet (up).
ORIGINAL	
ADD 8 TO X	Reset Bob's x position.
LET B 4	
ENDIF	
ENDIF	
IF DIRECTION = 2	If Bob is moving right, and

IF CANGORIGHT	if there is space to the right of Bob, then fire away...
ADD 8 TO Y	Reposition for bullet.
SPAWN 1 4	Type 1, Image 4 (Bullet)
SPAWNED	
LET FRAME 1	Set the frame for this bullet (right).
LET DIRECTION 2	Set the direction for this bullet (right).
ORIGINAL	
SUBTRACT 8 TO Y	Reset Bob's y position.
LET B 4	
ENDIF	
ENDIF	
IF DIRECTION = 3	If Bob is moving down, and
IF CANGODOWN	if there is space to below Bob, then
ADD 8 TO X	Reposition for bullet.
SPAWN 1 4	Type 1, Image 4 (Bullet)
SPAWNED	
LET FRAME 2	Set the frame for this bullet (down).
LET DIRECTION 3	Set the direction for this bullet (down).
ORIGINAL	
SUBTRACT 8 TO X	Reset Bob's x position.
LET B 4	
ENDIF	
ENDIF	
IF DIRECTION = 4	If Bob is moving left, and
IF CANGOLEFT	if there is space to the left of Bob, then
SUBTRACT 8 TO Y	Reposition for bullet.
SPAWN 1 4	Type 1, Image 4 (Bullet)
SPAWNED	
LET FRAME 3	Set the frame for this bullet (left).
LET DIRECTION 4	Set the direction for this bullet (left).
ORIGINAL	
ADD 8 TO Y	Reset Bob's y position.
LET B 4	
ENDIF	
ENDIF	
ENDIF	
IF B = 4	
BEEP 40	
SUBTRACT 1 TO PARAMA	
LET K PARAMA	
ENDIF	
SPRITEINK 70	Bright YELLOW
IF A = 0	If it is time to animate,
IF R = 1	then animate Bob...
ANIMATE	
ENDIF	
ENDIF	
ENDIF	...done animating.
LET M X	Store Bob's X and Y positions to allow
LET N Y	the zombies to track Bob.
ENDIF	

Type 1 - Bullet

SPRITEINK 4 GREEN

LET PARAMB 0	Reset the remove bullet indicator.
IF COLLISION 2	If this bullet has hit a zombie, then
OTHER	
LET RND = IMAGE	Get the current zombie sprite image
ORIGINAL	
IF RND <> 9	If this zombie isn't currently dying,
OTHER	then kill it...
LET IMAGE 9	Exploding head and melt down sprite.
LET FRAME 0	
ORIGINAL	
SCORE 1	Add 10 to the score.
COLOUR 79	Update the on screen score...
LET LINE 1	
LET COLUMN 24	
SHOWSCORE	..done.
LET PARAMB 1	Remove the bullet.
ENDIF	
ENDIF	
IF PARAMB = 0	If the bullet is to stay, then
IF J > 0	If the screen is redrawing, then
LET PARAMB 1	remove the bullet.
ENDIF	
IF DIRECTION = 1	If going up, and
IF CANGOUP	the sprite can go up, then
SUBTRACT 4 TO X	move it up 4 pixels.
ELSE	
LET PARAMB 1	Otherwise, remove it.
ENDIF	
ENDIF	
IF DIRECTION = 2	If going right, and
IF CANGORIGHT	the sprite can go right, then
ADD 4 TO Y	move it right 4 pixels.
ELSE	
LET PARAMB 1	Otherwise, remove it.
ENDIF	
ENDIF	
IF DIRECTION = 3	If going down, and
IF CANGODOWN	the sprite can go down, then
ADD 4 TO X	move it down 4 pixels.
ELSE	
LET PARAMB 1	Otherwise, remove it.
ENDIF	
ENDIF	
IF DIRECTION = 4	If going left, and
IF CANGOLEFT	the sprite can go right, then
SUBTRACT 4 TO Y	move it right 4 pixels.
ELSE	
LET PARAMB 1	
ENDIF	Otherwise, remove it.
ENDIF	
IF Y > 238	If at the edge of the screen...
LET PARAMB 1	...remove it...
ENDIF	
IF Y <= 0	
LET PARAMB 1	...remove it...
ENDIF	

IF X <= 16	
LET PARAMB 1	...remove it...
ENDIF	
IF X > 176	
LET PARAMB 1	...remove it...
ENDIF	...done edge check.
ENDIF	
IF PARAMB = 1	If the bullet is to be removed, then...
SUBTRACT 1 TO C	Decrement the sprite counter.
SUBTRACT 1 TO D	Decrement the bullet counter.
REMOVE	Remove the sprite.
ENDIF	...done.

Type 2 - Zombie	
IF IMAGE = 9	If the zombie is “dying”, then
IF FRAME = 3	if this is frame 3 (head exploding), then
IF PARAMA = 0	
IF C <= 11	if there is a free sprite, then
LET PARAMA 1	
LET RND PARAMB	Store this zombie’s colour.
IF DIRECTION = 1	This section determines which side to
IF CANGOUP	fire off the mutated pathogen...
SUBTRACT 8 TO X	
SPAWN 5 11	
SPAWNED	
LET FRAME 0	
LET DIRECTION 1	
LET PARAMA RND	Assign same colour as the zombie.
ORIGINAL	
ADD 8 TO X	
ENDIF	
ENDIF	
IF DIRECTION = 2	
IF CANGORIGHT	
ADD 8 TO Y	
SPAWN 5 11	
LET FRAME 0	
LET DIRECTION 2	
LET PARAMA RND	Assign same colour as the zombie.
ORIGINAL	
SUBTRACT 8 TO Y	
ENDIF	
ENDIF	
IF DIRECTION = 3	
IF CANGODOWN	
ADD 8 TO X	
SPAWN 5 11	
SPAWNED	
LET FRAME 0	
LET DIRECTION 3	
LET PARAMA RND	Assign same colour as the zombie.
ORIGINAL	
SUBTRACT 8 TO X	
ENDIF	
ENDIF	
ENDIF	

IF DIRECTION = 4	
IF CANGOLEFT	
SUBTRACT 8 TO Y	
SPAWN 5 11	
SPAWNED	
LET FRAME 0	
LET DIRECTION 4	
LET PARAMA RND	Assign same colour as the zombie.
ORIGINAL	
ADD 8 TO Y	
ENDIF	
ENDIF	...done with the mutated pathogen.
ENDIF	
ENDIF	
ENDIF	
ENDIF	
SPRITEINK PARAMB	Assigned during sprite initialisation.
IF OPT = 0	If ok to move robot (movement counter),
IF IMAGE <> 9	If not dying...
LET DIRECTION 0	Reset direction indicator.
IF M > X	This section is a simple tracker for
IF CANGODOWN	the player...
SPRITEDOWN	
LET DIRECTION 3	
ENDIF	
ELSE	
IF M <> X	
IF CANGOUP	
SPRITEUP	
LET DIRECTION 1	
ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF N > Y	
IF CANGORIGHT	
SPRITERIGHT	
LET DIRECTION 2	
IF IMAGE <> 5	Right facing zombie.
LET IMAGE 5	
LET FRAME 0	
ENDIF	
ENDIF	
ELSE	
IF N <> Y	
IF CANGOLEFT	
SPRITELEFT	
LET DIRECTION 4	
IF IMAGE <> 6	
LET IMAGE 6	Left facing zombie.
LET FRAME 0	
ENDIF	
ENDIF	
ENDIF	
ENDIF	...done tracking the player.
ELSE	
IF FRAME = 2	If the head is exploding...

BEEP 30	BEEP!
ENDIF	
IF FRAME = 9	If this is the end of the animation, then
REMOVE	remove the sprite, and
SUBTRACT 1 TO C	decrement the sprite counter.
ENDIF	
ENDIF	
IF DIRECTION > 0	If moving, then
ANIMATE	animate.
ELSE	
LET FRAME 0	Otherwise, stand still.
ENDIF	
ENDIF	
IF COLLISION 0	
IF IMAGE <> 9	If the zombie is not already dying,
BEEP 60	
LET IMAGE 9	Set zombie to die.
LET FRAME 0	
OTHER	
LET RND IMAGE	
IF RND <> 7	
LET IMAGE 7	Set Bob to die.
LET FRAME 0	
SUBTRACT 1 TO LIVES	Decrement lives counter.
LET B 3	Update onscreen lives counter.
ENDIF	
ORIGINAL	
ENDIF	
ENDIF	

Type 3 - Ammo Box	
SPRITEINK 87	
IF OPT = 0	If it's ok to animate, then
ANIMATE	animate.
ENDIF	
IF COLLISION 0	
BEEP 60	
OTHER	
IF PARAMB <= 93	If the player has less than 93 extra bullets, then
ADD 6 TO PARAMB	add another 6.
ENDIF	
LET K PARAMB	
ORIGINAL	
REMOVE	Remove the ammo box.
SUBTRACT 1 TO C	Decrement the sprite counter.
LET LINE 23	
LET COLUMN 21	
COLOUR 87	
MESSAGE 13	"Bullet x 00"
IF PARAMB <= 9	
LET COLUMN 24	
ELSE	
LET COLUMN 23	
ENDIF	
LET LINE 23	

DISPLAY K	Display the number of extra bullets.
ENDIF	

Type 4 - Heart	
SPRITEINK 2	RED
IF A = 0	If it's time to animate, then
LET FRAME 0	Reset current frame (smaller heart).
IF PARAMB = 10	If the frame counter is 10, then
LET FRAME 1	Set frame to 1 (larger heart), and
BEEP 2	BEEP!
ENDIF	
IF PARAMB = 12	If the frame counter is 12, then
LET FRAME 1	Set frame to 1 (larger heart), and
BEEP 2	BEEP!
ENDIF	
ADD 1 TO PARAMB	Increment the frame counter
IF PARAMB > 12	We only want 12 frames in the sequence,
LET PARAMB 0	so reset if more than 12.
ENDIF	
ENDIF	
IF COLLISION 0	
BEEP 60	
ADD 1 TO LIVES	Increment lives counter.
LET B 3	Display lives on screen.
REMOVE	Remove the heart.
SUBTRACT 1 TO C	Decrement the sprite counter.
ENDIF	

Type 5 - Virus (Zombie bullet)	
IF A = 0	If it's time to animate, then
ANIMATE	animate.
ENDIF	
SPRITEINK PARAMA	Colour from parent zombie.
LET PARAMB 0	Reset sprite removal indicator.
IF COLLISION 0	If collided with the player, then
BEEP 60	
OTHER	
IF IMAGE <> 7	If the player is not already dying, then
LET IMAGE 7	kill the player.
LET FRAME 0	
SUBTRACT 1 TO LIVES	Decrement lives counter.
ORIGINAL	
LET B 3	Display lives on screen.
ENDIF	
LET PARAMB 1	Remove the virus.
ENDIF	
IF COLLISION 2	If collided with a zombie, then
OTHER	
LET RND = IMAGE	Get the image of the zombie.
ORIGINAL	
IF RND <> 9	If the zombie is not already "dying",
OTHER	
LET IMAGE 9	Kill the zombie.
LET FRAME 0	

ORIGINAL	
SCORE 1	Add 10 to the score.
COLOUR 79	
LET LINE 1	
LET COLUMN 24	
SHOWSCORE	Update the onscreen score.
ENDIF	
ENDIF	
IF PARAMB = 0	If the virus is still in play, then
IF J > 0	if the screen is redrawing,
LET PARAMB 1	remove the sprite.
ENDIF	
IF DIRECTION = 1	If going up, and
IF CANGOUP	the sprite can go up, then
SUBTRACT 3 TO X	move it up 3 pixels.
ELSE	
LET PARAMB 1	Otherwise, remove it.
ENDIF	
ENDIF	
IF DIRECTION = 2	If going right, and
IF CANGORIGHT	the sprite can go right, then
ADD 3 TO Y	move it right 3 pixels.
ELSE	
LET PARAMB 1	Otherwise, remove it.
ENDIF	
ENDIF	
IF DIRECTION = 3	If going down, and
IF CANGODOWN	the sprite can go down, then
ADD 3 TO X	move it down 3 pixels.
ELSE	
LET PARAMB 1	Otherwise, remove it.
ENDIF	
ENDIF	
IF DIRECTION = 4	If going left, and
IF CANGOLEFT	the sprite can go right, then
SUBTRACT 3 TO Y	move it right 3 pixels.
ELSE	
LET PRAMAB 1	
ENDIF	Otherwise, remove it.
ENDIF	
IF Y > 238	If at the edge of the screen...
LET PARAMB 1	...remove it...
ENDIF	
IF Y <= 0	
LET PARAMB 1	...remove it...
ENDIF	
IF X <= 16	
LET PARAMB 1	...remove it...
ENDIF	
IF X > 176	
LET PARAMB 1	...remove it...
ENDIF	...done edge check.
ENDIF	
IF PARAMB = 1	If the virus is to be removed, then...
SUBTRACT 1 TO C	Decrement the sprite counter.
REMOVE	Remove the sprite.

ENDIF	...done.
Initialise sprite	
IF TYPE = 0	
LET PARAMA 0	Reset player variables...
LET PARAMB 0	...done.
ELSE	
ADD 1 TO C	Increment sprite counter.
IF TYPE = 1	If this is a bullet, then
ADD 1 TO D	increment the bullet counter.
ENDIF	
IF TYPE = 2	If this sprite is a zombie,
GETRANDOM 2	Decide to remove or keep the zombie.
IF RND = 1	If indicator is 1, then
REMOVE	remove the sprite, and
SUBTRACT 1 TO C	decrement sprite counter.
ELSE	Otherwise, keep the sprite.
GETRANDOM 2	Add a bit of variety
LET FRAME RND	to zombie animation.
GET RANDOM 6	Decide what colour to use
ADD 1 TO RND	Colours need to be between 1 and 7
IF RND = 6	If YELLOW (Bob's colour), then
LET RND = 7	change to WHITE.
ENDIF	
ADD 64 TO RND	Make the colour BRIGHT.
LET PARAMB RND	Store the sprite's colour.
LET PARAMA 0	
ENDIF	
ENDIF	
ENDIF	

Main loop 1	
IF SCREEN = 1	If this is the title screen, then
IF A = 0	If it's time to animate, then
IF OPT = 0	again, if it's time to animate, then
IF 0 = 0	if this is the first time here, then
LET O 1	initialise variables. These control the
LET C 32	scrolling text.
LET R 3	
LET Q C	
COLOUR 69	
LET LINE 18	
LET COLUMN 1	
MESSAGE 33	"Q=Up, A=Down, O=Left, P=Right"
LET LINE 19	
LET COLUMN 1	
MESSAGE 34	"SPACE=Fire, R=Reload, H=Pause"
COLOUR 71	
LET LINE 20	
LET COLUMN 5	
MESSAGE 35	"Press SPACE to start"
ENDIF	
COLOUR 7	
REPEAT 12	12 visible lines of text. This section

LET LINE R	displays the scrolling text on the title
LET COLUMN 14	screen...
IF Q <= 31	
MESSAGE Q	
ELSE	
MESSAGE 32	Blank line.
ENDIF	
ADD 1 TO Q	
IF Q > 43	
LET Q 17	
ENDIF	
ADD 1 TO R	
IF R > 14	
LET R 3	
ADD 1 TO C	
IF C > 43	
LET C 17	
ENDIF	
LET Q C	
ENDIF	
ENDREPEAT	...done scrolling text.
ENDIF	
ENDIF	
LET B 0	Reset B so nothing else happens.
IF KEY 4	If the start button ("SPACE") is pressed,
LET SCREEN 0	change to the game screen, and...
COLOUR 87	
LET LINE 23	
LET COLUMN 0	
MESSAGE 9	...clear the copyright message, and
LET O 1	reset garden maze co-ordinates, and
LET P 1	
LET J 253	tell player event to reset Bob, etc.
EXIT	exit immediately.
ENDIF	
ENDIF	
IF A = 0	Animate flip-flop...
ADD 1 TO A	
ELSE	
LET A 0	
ENDIF	...done.
IF OPT <= 3	Zombie animation indicator...
ADD 1 TO OPT	
ELSE	
LET OPT 0	
ENDIF	...done.
IF B > 0	B is a display indicator which does much:
IF B = 4	Update bullet indicators:
LET RND 6	Temp used bullet counter.
SUBTRACT K TO RND	Subtract number of bullets left.
COLOUR 80	BLACK on RED.
LET Q 26	
REPEAT RND	Display empty bullet for each bullet
LET LINE 23	used...
LET COLUMN Q	
MESSAGE 12	Bullet.

ADD 1 TO Q	
ENDREPEAT	...done showing empty bullets.
ENDIF	
IF B = 3	Update lives indicators...
COLOUR 79	
LET LINE 1	
LET COLUMN LIVES	
MESSAGE 2	2 spaces.
IF LIVES > 0	
LET RND 0	
REPEAT LIVES	
LET COLUMN RND	
LET LINE 1	
MESSAGE 6	Mini Bob.
ADD 1 TO RND	
ENDREPEAT	
ENDIF	
LET B 1	
ENDIF	...done showing lives.
IF B = 1	Update garden maze section co-ordinates..
IF 0 > 0	
COLOUR 87	
LET LINE 23	
LET COLUMN 0	
MESSAGE 5	“ , “
LET LINE 23	
IF 0 <= 9	
LET COLUMN 2	
ELSE	
IF 0 <= 99	
LET COLUMN 1	
ELSE	
LET COLUMN 0	
ENDIF	
ENDIF	
DISPLAY 0	
LET LINE 23	
IF P <= 9	
LET COLUMN 6	
ELSE	
IF P <= 99	
LET COLUMN 5	
ELSE	
LET COLUMN 4	
ENDIF	
ENDIF	
DISPLAY P	
ENDIF	
ENDIF	...done showing co-ordinates.
LET B 0	Reset indicator so nothing else happens.
ENDIF	
Main loop 2	
IF J = 254	
LET LIVES 3	

LET D 0	
LET E 0	
LET OPT 4	
LET C 0	
LET K 0	
LET L 0	
LET M 0	
LET N 0	
LET J 10	
EXIT	
ENDIF	
IF J > 200	
EXIT	
ENDIF	
IF J > 0	If the screen is in redraw mode, and
IF J > 9	If the indicator is > 9 (ie. not a peg),
IF J = 10	and redrawing has not yet started, and
IF E > 0	there were some sprites on the screen,
IF C = 0	and all have been removed, then
COLOUR 120	Display the "clearance bonus" message...
LET LINE 9	
LET COLUMN 8	
MESSAGE 16	Blank line.
LET LINE 10	
LET COLUMN 8	
MESSAGE 14	"Clearance Bonus"
LET LINE 11	
LET COLUMN 8	
MESSAGE 15	"? X 100"
LET LINE 11	
LET COLUMN 13	
DISPLAY E	
LET LINE 12	
LET COLUMN 8	
MESSAGE 16	Blank line.
DELAY 40	
COLOUR 79	
REPEAT E	
BEEP 60	
SCORE 1	
LET LINE 1	
LET COLUMN 24	
SHOWSCORE	
DELAY 10	
ENDREPEAT	
DELAY 40	
ENDIF	...done with the clearance bonus.
ENDIF	
LET SCREEN 0	Reset the screen.
ENDIF	
SUBTRACT 1 TO J	Decrement the "peg" counter.
IF J = 9	
LET B 1	Update garden maze co-ordinates.
ENDIF	
ELSE	
COLOUR 4	GREEN on BLACK.

IF M = 26	These sections basically block the exit
LET E 2	so that you are forced to go through the
LET F 13	next section and cannot quickly escape
REPEAT 6	a dire situation...
LET LINE E	
LET COLUMN F	
PUTBLOCK 1	1 hedge block.
LET LINE E	
LET COLUMN F	
MESSAGE 1	1 hedge block.
ADD 1 TO F	
ENDREPEAT	
ENDIF	
IF M = 160	
LET E 22	
LET F 13	
REPEAT 6	
LET LINE E	
LET COLUMN F	
PUTBLOCK 1	1 hedge block.
LET LINE E	
LET COLUMN F	
MESSAGE 1	1 hedge block.
ADD 1 TO F	
ENDREPEAT	
ENDIF	
IF N = 10	
LET E 10	
LET F 0	
REPEAT 5	
LET LINE E	
LET COLUMN F	
PUTBLOCK 1	1 hedge block.
LET LINE E	
LET COLUMN F	
MESSAGE 1	1 hedge block.
ADD 1 TO E	
ENDREPEAT	
ENDIF	
IF N = 228	
LET E 10	
LET F 31	
REPEAT 5	
LET LINE E	
LET COLUMN F	
PUTBLOCK 1	1 hedge block.
LET LINE E	
LET COLUMN F	
MESSAGE 1	1 hedge block.
ADD 1 TO E	
ENDREPEAT	
ENDIF	...done blocking exits.
LET J 1	Reset the current "peg" counter.
REPEAT 8	Repeat for each of the 8 "pegs".
LET K 0	These lines are the formula used to
ADD P TO K	calculate the actual maze walls for each

LET L 0	section of the maze. This is what allows
ADD P TO L	over 16,000 (mostly) unique screens.
MULTIPLY L BY J	
MULTIPLY L BY 3	
ADD L TO K	
IF K > 4	I could have just done a mod equivalent
DIVIDE K BY 3	here I suppose, but, what-the-hell...
ENDIF	
IF K > 4	
DIVIDE K BY 3	
ENDIF	
IF K > 4	
DIVIDE K BY 3	
ENDIF	
IF K > 4	
DIVIDE K BY 3	
ENDIF	
IF K > 4	
DIVIDE K BY 3	
ENDIF	
IF K > 4	
DIVIDE K BY 3	
ENDIF	
IF J = 1	...done not using mod.
LET E K	Store the direction of each "peg" in
ENDIF	its associated variable for use later
IF J = 2	on...
LET F K	
ENDIF	
IF J = 3	
LET G K	
ENDIF	
IF J = 4	
LET H K	
ENDIF	
IF J = 5	
LET I K	
ENDIF	
IF J = 6	
LET L K	
ENDIF	
IF J = 7	
LET Q K	
ENDIF	
IF J = 8	
LET R K	
ENDIF	...all "pegs" stored.
ADD 1 TO J	Increment the "peg" counter.
ENDREPEAT	Repeat until all "pegs" are done.
LET K 0	This section basically does a modulus
LET J K	division to find out the remainder. This
DIVIDE K BY 8	remainder is then used in the next
MULTIPLY K BY 8	section to "rotate" the "pegs" to reduce
SUBTRACT K FROM J	the chances of duplicate screens.
REPEAT J	
LET K E	This section rotates the "pegs" to reduce
LET E F	the chances of duplicate screens which
LET F G	was an issue in the original Berzerk
LET G H	game. Duplicates may still occur, but

LET H I	this should reduce the likelihood quite
LET I L	substantially.
LET L Q	
LET Q R	
LET R K	
ENDREPEAT	...done rotating.
LET J 1	Reset "peg" counter.
REPEAT 8	Repeat for all 8 "pegs".
IF J <= 4	
LET M 9	Top row of pegs.
ELSE	
LET M 15	Bottom row of pegs.
ENDIF	
IF J = 1	
LET N 6	
LET K E	
ENDIF	
IF J = 2	
LET N 12	
LET K F	
ENDIF	
IF J = 3	
LET N 19	
LET K G	
ENDIF	
IF J = 4	
LET N 25	
LET K H	
ENDIF	
IF J = 5	
LET N 6	
LET K I	
ENDIF	
IF J = 6	
LET N 12	
LET K L	
ENDIF	
IF J = 7	
LET N 19	
LET K Q	
ENDIF	
IF J = 8	
LET N 25	
LET R	
ENDIF	
COLOUR 4	GREEN
IF K = 1	Up from the "peg".
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	

PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ENDIF	
IF K = 2	Right from the "peg".
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	

LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ENDIF	
IF K = 3	Down from the "peg".
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO M	

LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ADD 1 TO M	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ENDIF	
IF K = 4	Left from the “peg”.
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.

SUBTRACT 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
SUBTRACT 1 TO N	
LET LINE M	
LET COLUMN N	
PUTBLOCK 1	1 hedge block.
LET LINE M	
LET COLUMN N	
MESSAGE 1	1 hedge block.
ENDIF	
ADD 1 TO J	Increment "peg" counter.
ENDREPEAT	Repeat for each "peg".
LET J 255	Turn off "peg" processing.
ENDIF	
ENDIF	

Game initialisation	
LET LIVES 3	
LET D 0	
LET OPT 4	
LET C 0	
LET O 0	
LET P 0	
LET J 1	
LET K 0	
LET L 0	
LET M 0	
LET N 0	
BORDER 0	
COLOUR 71	
CLS	
COLOUR 78	
LET LINE 0	
LET COLUMN 0	
MESSAGE 10	"The Adventures of Bouncing Bob"
COLOUR 77	
LET LINE 1	
LET COLUMN 0	
MESSAGE 9	Blank line.
LET LINE 1	
LET COLUMN 9	
MESSAGE 3	"Lurching Dead"
MESSAGE 8	

COLOUR 87	
LET LINE 23	
LET COLUMN 0	
MESSAGE 9	Blank line.
LET B 1	
LET LINE 23	
LET COLUMN 6	
MESSAGE 4	"©2015 retrific.com"

Restart screen	
LET C 0	Reset sprite counter.
LET D 0	Reset bullet counter.
LET B 3	
LET E 0	Reset clearance bonus counter.