

The Adventures of Bouncing Bob

Summer Breeze

By Douglas Bagnall

©2015 retrific.com

[www.retrific.com](http://www.retrific.com)

## Table of Contents

Events.....	1
Player control (type 0).....	1
Sprite type 1 (Evil Bob) .....	7
Sprite type 2 (Bat) .....	7
Sprite type 3 (Spider).....	8
Sprite type 4 (Frog Head) .....	8
Sprite type 5 (Collectable Objects) .....	9
Sprite type 7 (Teleporter).....	10
Sprite type 8 (Extraction Fan).....	11
Initialise sprite.....	12
Main loop 1.....	14
Main loop 2.....	16
Game initialisation .....	19
Restart screen.....	19
Fell too far.....	20
Kill player.....	20
Lost game .....	20
Completed game .....	20
Global Variables (A to P).....	20
Miscellaneous.....	21
Collision distance.....	21
Sprite height.....	21
SPRITEINK mask .....	21
Order sprites.....	21
Specialise.....	21
AY sounds .....	21
Objects.....	21
Sprites.....	21
Text .....	23

## Events

### Player control (type 0)

IF P = 0	Is the game in dying mode (i.e. player dying)?
IF IMAGE = 0	Is the current sprite image the default player image?
LET IMAGE 3	Change the current sprite image to the dying player.
ENDIF	
IF CANGODOWN	Can the player sprite move downwards?
FALL	Cause the player sprite to move down.
ELSE	If the player sprite cannot move downwards:
IF A = 0	Should the sprite be animated? If so,
SPRITEINK 6	set the sprite ink to yellow.
ANIMATE	Animate the sprite (show the next frame of the sprite).
IF FRAME = 0	If the sprite has fully animated and started again:
LET P 1	Set the game to play mode.
LET IMAGE 0	change the current sprite to the default image.
EXIT	Exit immediately.
ENDIF	
BEEP 4	Beep!
ENDIF	
ENDIF	
ELSE	Else, if the game is in play mode:
IF SCREEN = 1	Is this screen 1?
IF 0 = 0	If nothing else is going on,
LET 0 = 8	then show the batteries and/or messages.
ENDIF	
ENDIF	
IF LIVES = 0	Has the player used all of their lives? If so,
KILL	run the kill player script.
ENDIF	
IF LIVES <= 2	Is the player running out of lives?
IF 0 = 0	If nothing else is happening
LET PARAMA X	Store player x position
LET PARAMB Y	Store player Y position
IF SCREEN = 12	Is the player on screen 12? Is so,
IF GOT 4	has the player already picked up the heart? If so,
LET X 96	
LET Y 200	
PUT 4	replace the heart so it can be reused.
SPAWN 5 8	Show the heart sprite.
ENDIF	
ENDIF	
IF SCREEN = 30	Is the player on screen 30? Is so,
IF GOT 5	has the player already picked up the heart? If so,
LET X 152	
LET Y 176	
PUT 5	replace the heart so it can be reused.
SPAWN 5 8	Show the heart sprite.
ENDIF	
ENDIF	
IF SCREEN = 7	Is the player on screen 7? Is so,
IF GOT 6	has the player already picked up the heart? If so,
LET X 128	
LET Y 184	
PUT 6	replace the heart so it can be reused.
SPAWN 5 8	Show the heart sprite.
ENDIF	
ENDIF	
IF SCREEN = 18	Is the player on screen 18? Is so,
IF GOT 9	has the player already picked up the heart? If so,
LET X 32	
LET Y 224	
PUT 9	replace the heart so it can be reused.

SPAWN 5 8	Show the heart sprite.
ENDIF	
ENDIF	
IF SCREEN = 26	Is the player on screen 26? Is so,
IF GOT 10	has the player already picked up the heart? If so,
LET X 32	
LET Y 16	
PUT 10	replace the heart so it can be reused.
SPAWN 5 8	Show the heart sprite.
ENDIF	
ENDIF	
LET X PARAMA	Restore player X position.
LET Y PARAMB	Restore player Y position.
ENDIF	
ENDIF	
IF CUSTOM	Has the player hit a custom block (teleporter)?
IF SCREEN = 22	Is the player on screen 22?
LET SCREEN 5	Jump to screen 5.
LET X 88	Position the player.
LET Y 16	Position the player.
ENDIF	
IF SCREEN = 30	Is the player on screen 30?
LET SCREEN 12	Jump to screen 12.
LET X 28	Position the player.
LET Y 120	Position the player.
ENDIF	
ENDIF	
IF SCREEN = 7	Is the player on screen 7?
IF GOT 3	Has the player got object 3 (Front Door Key)?
IF L = 0	If the door has not been removed:
LET L 1	Indicator so that the door is not constantly removed.
COLOUR 70	Paper 0 (Black), Ink 6 (Yellow), BRIGHT.
LET LINE 19	The following code removes the front door:
LET COLUMN 15	
PUTBLOCK 0	Empty space to allow player to walk through the door.
LET LINE 19	In AGD 4.6, there is an error which corrupts blocks
LET COLUMN 15	when the text is displayed. This message hides
MESSAGE 9	the corrupt block with a space.
LET LINE 20	
LET COLUMN 15	
PUTBLOCK 0	
LET LINE 20	
LET COLUMN 15	
MESSAGE 9	As above.
LET LINE 21	
LET COLUMN 15	
PUTBLOCK 0	
LET LINE 21	
LET COLUMN 15	
MESSAGE 9	As above.
ENDIF	
ENDIF	
IF Y > 18	Is the player within the bounds of the screen?
IF Y <= 228	Note: X and Y are swapped in AGD.
IF GOT 3	Has the player got object 3 (Front Door Key)?
ELSE	If not,
IF M <> SCREEN	if not already displaying the message,
LET M 107	set to display message 7
ENDIF	
ENDIF	
ELSE	If player is about to leave the screen to the right,
LET M 98	hide any message that may be displayed
ENDIF	
ENDIF	
IF Y <= 18	If the player is about to leave the screen to the left,

LET M 98	hide any message that may be displayed
ENDIF	
ENDIF	
IF SCREEN = 10	Is the player on screen 10?
IF GOT 2	Has the player got object 2 (Balcony Door Key)?
IF L = 0	If the door has not been removed:
LET L 1	Indicator so that the door is not constantly removed.
COLOUR 70	Paper 0 (Black), Ink 6 (Yellow), BRIGHT.
LET LINE 19	The following code removes the Balcony door:
LET COLUMN 30	
PUTBLOCK 0	Empty space to allow player to walk through the door.
LET LINE 19	In AGD 4.6, there is an error which corrupts blocks
LET COLUMN 30	when the text is displayed. This message hides
MESSAGE 9	the corrupt block with a space.
LET LINE 20	
LET COLUMN 30	
PUTBLOCK 0	
LET LINE 20	
LET COLUMN 30	
MESSAGE 9	As above.
LET LINE 21	
LET COLUMN 30	
PUTBLOCK 0	
LET LINE 21	
LET COLUMN 30	
MESSAGE 9	As above.
ENDIF	
ENDIF	
IF Y > 18	Is the player within the bounds of the screen?
IF GOT 2	Has the player got object 2 (Balcony Door Key)?
ELSE	If not,
IF M <> SCREEN	if not already displaying the message,
LET M 106	set to display message 6
ENDIF	
ENDIF	
ENDIF	
IF Y <= 17	If player is about to leave the screen to the left,
IF GOT 2	
ELSE	If the player does not have object 2,
LET M 98	hide any message that may be displayed
ENDIF	
ENDIF	
ENDIF	
IF SCREEN = 0	Is the player on screen 0?
IF L = 0	If the doors have not been removed:
LET L 1	Indicator so that the door is not constantly removed.
IF GOT 0	Has the player got object 0 (Right Basement Door Key)?
COLOUR 70	Paper 0 (Black), Ink 6 (Yellow), BRIGHT.
LET LINE 19	The following code removes the Balcony door:
LET COLUMN 30	
PUTBLOCK 0	Empty space to allow player to walk through the door.
LET LINE 19	In AGD 4.6, there is an error which corrupts blocks
LET COLUMN 30	when the text is displayed. This message hides
MESSAGE 9	the corrupt block with a space.
LET LINE 20	
LET COLUMN 30	
PUTBLOCK 0	
LET LINE 20	
LET COLUMN 30	
MESSAGE 9	As above.
LET LINE 21	
LET COLUMN 30	
PUTBLOCK 0	
LET LINE 21	
LET COLUMN 30	

MESSAGE 9	As above.
ENDIF	
IF GOT 1	Has the player got object 1 (Left Basement Door Key)?
COLOUR 70	Paper 0 (Black), Ink 6 (Yellow), BRIGHT.
LET LINE 19	The following code removes the Balcony door:
LET COLUMN 1	
PUTBLOCK 0	Empty space to allow player to walk through the door.
LET LINE 19	In AGD 4.6, there is an error which corrupts blocks
LET COLUMN 1	when the text is displayed. This message hides
MESSAGE 9	the corrupt block with a space.
LET LINE 20	
LET COLUMN 1	
PUTBLOCK 0	
LET LINE 20	
LET COLUMN 1	
MESSAGE 9	As above.
LET LINE 21	
LET COLUMN 1	
PUTBLOCK 0	
LET LINE 21	
LET COLUMN 1	
MESSAGE 9	As above.
ENDIF	
ENDIF	
IF X = 160	Is the player on the bottom floor?
IF Y <= 84	Is the player near the left basement door?
IF M <> SCREEN	if not already displaying the message,
IF GOT 1	and the player does not have the key to the door,
ELSE	
LET M 105	then display message 5.
ENDIF	
ENDIF	
ENDIF	
IF Y > 173	Is the player near the right basement door?
IF M <> SCREEN	if not already displaying the message,
IF GOT 0	and the player does not have the key to the door,
ELSE	
LET M 104	then display message 4.
ENDIF	
ENDIF	
ENDIF	
IF Y > 84	If the player is not near any basement door,
IF Y <= 173	
LET M 98	then hide any message which may be displayed.
ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF M > 100	Is there a message to be displayed? If so,
SUBTRACT 100 FROM M	subtract 100 from M to find the actual message number,
LET LINE 23	then position the text.
LET COLUMN 1	
COLOUR 7	Paper 0 (Black), Ink 7 (White)
MESSAGE M	
LET M SCREEN	The message for the current screen is being displayed.
LET N 1	There is a message which can be cleared when required.
ENDIF	
IF M = 98	Is M to "clear message area" (98)?
IF N = 1	Is there a message to be cleared?
COLOUR 1	Paper 0 (Black), Ink 1 (Blue)
LET LINE 23	Position the text.
LET COLUMN 1	
MESSAGE 11	Display the copyright message.
LET M 99	No message to be displayed.
LET N 0	Message area already cleared, no action required.

ENDIF	
ENDIF	
DETECTOBJ	
IF OBJ <> 255	Has the player made contact with an object? If so,
IF OBJ > 17	Is this object a switch?
IF 0 = 0	If nothing is currently happening:
LET 0 9	Indicate that a switch is being used.
LET PARAMA X	Store player x position
LET PARAMB Y	Store player y position
LET X 136	
LET Y 168	
IF GOT 18	Has the player got the boiler ON switch?
IF OBJ = 19	If so, then check for the boiler OFF switch.
GET OBJ	Get the OFF switch.
PUT 18	Replace it on screen with the ON switch.
ENDIF	
ELSE	
IF GOT 19	Has the player got the boiler OFF switch?
IF OBJ = 18	If so, check for the boiler ON switch.
GET OBJ	Get the ON switch.
PUT 19	Replace it on screen with the OFF switch.
ENDIF	
ENDIF	
ENDIF	
LET X PARAMA	Restore player x position
LET Y PARAMB	Restore player y position
ENDIF	
ELSE	
GET OBJ	pick up the object.
IF OBJ = 0	Is this the "Right Basement Key"? If so,
LET 0 3	flash the "Right Basement Key" icon.
ELSE	
IF OBJ <= 3	Is this another key? If so,
LET 0 OBJ	
ADD 3 TO 0	flash the relevant key icon.
ELSE	
IF OBJ <= 10	Is this a heart? If so,
LET 0 2	increment the LIVES counter, & flash the next life icon.
ELSE	
IF OBJ <= 17	Is this a battery? If so,
LET 0 7	increment the battery count, & flash the battery icon.
ENDIF	
ENDIF	
ENDIF	
ENDIF	
BEEP 30	Beep!
ENDIF	
IF A = 0	Should the sprite be animated? If so,
IF G = 0	is the player in normal mode? Is so,
SPRITEINK 6	set the sprite ink to yellow.
ELSE	else the player is in "God" mode, so
SUBTRACT 1 FROM G	Decrement the "God" counter by one.
IF F = 99	Skip screen change/sprite update issue.
SPRITEINK 6	
ELSE	
IF F > 2	Flash indicator - what colour to use?
SPRITEINK 6	Set the sprite ink to yellow.
ELSE	
SPRITEINK 0	Set the sprite ink to black.
ENDIF	
SUBTRACT 1 FROM F	Decrement the flash indicator.
IF F = 0	If the flash indicator is 0,
LET F 4	then reset it.
ENDIF	

ENDIF	
ENDIF	
ANIMATE	Animate the sprite (show the next frame of the sprite).
ENDIF	
IF X <= 26	If passed the top of the screen,
IF G = 0	If not in god mode
LET G 6	Set god mode for 6 cycles to correct for screen change/
LET F 99	sprite update issue.
ENDIF	
SCREENUP	move to the next screen
LET X 164	Reset the X position to the bottom of the screen
JUMP	Cause the player to jump
EXIT	Exit immediately.
ENDIF	
IF X > 166	If passed the bottom of the screen,
IF G = 0	If not in god mode
LET G 6	Set god mode for 6 cycles to correct for screen change/
LET F 99	sprite update issue.
ENDIF	
SCREENDOWN	move to the next screen.
LET X 28	Reset the X position to the top of the screen.
EXIT	Exit immediately.
ENDIF	
IF Y > = 232	If passed the right of the screen,
IF G = 0	If not in god mode
LET G 6	Set god mode for 6 cycles to correct for screen change/
LET F 99	sprite update issue.
ENDIF	
SCREENRIGHT	move to the next screen.
LET Y 10	Reset the Y position to the left of the screen.
SUBTRACT 1 FROM X	Reposition player one pixel up.
EXIT	Exit immediately.
ENDIF	
IF Y <= 8	If passed the left of the screen,
IF G = 0	If not in god mode
LET G 6	Set god mode for 6 cycles to correct for screen change/
LET F 99	sprite update issue.
ENDIF	
SCREENLEFT	move to the next screen.
LET Y 230	Reset the Y position to the right of the screen.
SUBTRACT 1 FROM X	Reposition player one pixel up.
EXIT	Exit immediately.
ENDIF	
IF KEY 0	If "right" key pressed,
IF CANGORIGHT	and player can move right,
SPRITERIGHT	move the player to the right.
ENDIF	
ENDIF	
IF KEY 1	If "left" key pressed,
IF CANGOLEFT	and player can move left,
SPRITELEFT	move the player to the left.
ENDIF	
ENDIF	
IF KEY 4	If "jump" key pressed,
JUMP	Jump.
BEEP 100	Beep!
ENDIF	
IF CANGODOWN	If there is nothing underneath the player,
FALL	fall!
ENDIF	
ENDIF	



## Sprite type 1 (Evil Bob)

IF P > 0	Is the game in play mode (i.e. Player not dying)?
IF PARAMA = 0	Is the sprite moving left?
IF CANGOLEFT	Can the sprite move left?
SPRITELEFT	Move the sprite left.
ELSE	If the sprite cannot move left,
LET PARAMA 1	set the sprite to move right.
ENDIF	
ELSE	
IF CANGORIGHT	Can the sprite move right?
SPRITERIGHT	Move the sprite right.
ELSE	If the sprite cannot move right,
LET PARAMA 0	set the sprite to move left.
ENDIF	
ENDIF	
SPRITEINK PARAMB	Colour defined in the Initialise Sprite event.
IF A = 0	Is it time to animate?
ANIMATE	Animate the sprite.
ENDIF	
IF G = 0	Is the player in normal mode (i.e. Not God mode)>
IF COLLISION 0	Has there been a collision with the player
LET 0 1	Decrement player's lives, and remove life icon.
ENDIF	
ENDIF	
ENDIF	

## Sprite type 2 (Bat)

IF P > 0	Is the game in play mode (i.e. Player not dying)?
IF PARAMA = 0	Is the sprite moving left?
IF CANGOLEFT	Can the sprite move left?
SPRITELEFT	Move the sprite left.
ELSE	If the sprite cannot move left,
LET PARAMA 1	set the sprite to move right.
ENDIF	
ELSE	If the sprite is moving right,
IF CANGORIGHT	Can the sprite move right?
SPRITERIGHT	Move the sprite right.
ELSE	If the sprite cannot move right,
LET PARAMA 0	set the sprite to move left.
ENDIF	
ENDIF	
IF DIRECTION = 0	Is the sprite moving up?
IF CANGROUP	Can the sprite move up?
SPRITEUP	Move the sprite up.
ELSE	If the sprite cannot move up,
LET DIRECTION 1	set the sprite to move down.
ENDIF	
ELSE	If the sprite is moving down,
IF CANGODOWN	Can the sprite move down?
SPRITEDOWN	Move the sprite down.
ELSE	If the sprite cannot move down,
LET DIRECTION 0	set the sprite to move up.
ENDIF	
ENDIF	
SPRITEINK 1	Set sprite ink to blue.
IF A = 0	Is it time to animate the sprite?
IF PARAMB <= 2	If the current frame sequence is <= 2,
LET FRAME PARAMB	set the sprite frame to the current sequence number.
ELSE	If the current frame sequence is > 2,
LET FRAME 1	set the frame to 1.
ENDIF	
ADD 1 TO PARAMB	Increment the frame sequence counter.

IF PARAMB > 3	If the frame sequence counter is > 3,
LET FRAME 0	reset the frame sequence counter to 0.
ENDIF	
ENDIF	
IF G = 0	Is the player in normal mode (i.e. Not God mode)>
IF COLLISION 0	Has there been a collision with the player
LET O 1	Decrement player's lives, and remove life icon.
ENDIF	
ENDIF	
ENDIF	

### Sprite type 3 (Spider)

IF P > 0	Is the game in play mode (i.e. Player not dying)?
IF PARAMA = 0	Is the sprite moving up?
IF CANGROUP	Can the sprite move up?
SPRITEUP	Move the sprite up.
ELSE	If the sprite cannot move up,
LET PARAMA 1	set the sprite to move down.
ENDIF	
ELSE	If the sprite is moving down,
IF CANGODOWN	Can the sprite move down?
SPRITEDOWN	Move the sprite down.
ELSE	If the sprite cannot move down,
LET PARAMA 0	set the sprite to move up.
ENDIF	
ENDIF	
SPRITEINK PARAMB	Colour defined in the Initialise Sprite event.
IF A = 0	Is it time to animate?
ANIMATE	Animate the sprite.
ENDIF	
IF G = 0	Is the player in normal mode (i.e. Not God mode)>
IF COLLISION 0	Has there been a collision with the player
LET O 1	Decrement player's lives, and remove life icon.
ENDIF	
ENDIF	
ENDIF	

### Sprite type 4 (Frog Head)

IF P > 0	Is the game in play mode (i.e. Player not dying)?
IF PARAMA = 0	Is the sprite moving left?
IF CANGOLEFT	Can the sprite move left?
SPRITELEFT	Move the sprite left.
IF IMAGE <> 7	If the sprite image is not facing left,
LET IMAGE 7	set the sprite image to face left.
ENDIF	
SUBTRACT 16 FROM Y	Check to see if the sprite is at the edge of a platform.
IF CANGODOWN	If the sprite is at the edge,
LET PARAMA 1	set the sprite to move right.
ENDIF	
ADD 16 TO Y	
ELSE	If the sprite cannot move left,
LET PARAMA 1	set the sprite to move right.
ENDIF	
ELSE	Is the sprite moving right?
IF CANGORIGHT	Can the sprite move right?
SPRITERIGHT	Move the sprite right.
IF IMAGE <> 6	If the sprite image is not facing right,
LET IMAGE 6	set the sprite image to face right.
ENDIF	
ADD 16 TO Y	Check to see if the sprite is at the edge of a platform.
IF CANGODOWN	If the sprite is at the edge,

LET PARAMA 0	set the sprite to move left.
ENDIF	
SUBTRACT 16 FROM Y	
ELSE	If the sprite cannot move right,
LET PARAMA 0	set the sprite to move left.
ENDIF	
ENDIF	
SPRITEINK 4	GREEN body.
SUBTRACT 8 FROM X	
SPRITEINK 6	YELLOW eyes.
ADD 8 TO X	
IF A = 0	Is it time to animate the sprite?
IF PARAMB <= 2	If the current frame sequence is <= 2,
LET FRAME PARAMB	set the sprite frame to the current sequence number.
ELSE	If the current frame sequence is > 2,
LET FRAME 1	set the frame to 1.
ENDIF	
ADD 1 TO PARAMB	Increment the frame sequence counter.
IF PARAMB > 3	If the frame sequence counter is > 3,
LET FRAME 0	reset the frame sequence counter to 0.
ENDIF	
ENDIF	
IF G = 0	Is the player in normal mode (i.e. Not God mode)>
IF COLLISION 0	Has there been a collision with the player
LET O 1	Decrement player's lives, and remove life icon.
ENDIF	
ENDIF	
ENDIF	

## Sprite type 5 (Collectable Objects)

IF IMAGE <> 8	If this is not a heart,
IF FRAME = 0	If the current frame is 0,
SPRITEINK 71	set the sprite ink to bright white
ELSE	Otherwise, if the current frame is not 0,
SPRITEINK 6	set the sprite ink to yellow
ENDIF	
ENDIF	
IF IMAGE = 12	If this is the front door key,
IF A = 0	Is it time to animate?
IF PARAMB = 0	If so, is the sprite being animated forward?
ANIMATE	If so, animate the sprite.
IF FRAME = 6	If this is the last frame,
LET PARAMB 1	set to animate backwards.
ENDIF	
ELSE	Is the sprite being animated backwards?
ANIMBACK	If so, animate the sprite backwards.
IF FRAME = 0	If this is the first frame,
LET PARAMB 0	set to animate forwards.
ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 11	If this is the balcony key,
IF A = 0	Is it time to animate?
IF PARAMB = 0	If so, is the sprite being animated forward?
ANIMATE	If so, animate the sprite.
IF FRAME = 6	If this is the last frame,
LET PARAMB 1	set to animate backwards.
ENDIF	
ELSE	Is the sprite being animated backwards?
ANIMBACK	If so, animate the sprite backwards.
IF FRAME = 0	If this is the first frame,
LET PARAMB 0	set to animate forwards.

ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 10	If this is the air con room key,
IF A = 0	Is it time to animate?
IF PARAMB = 0	If so, is the sprite being animated forward?
ANIMATE	If so, animate the sprite.
IF FRAME = 4	If this is the last frame,
LET PARAMB 1	set to animate backwards.
ENDIF	
ELSE	Is the sprite being animated backwards?
ANIMBACK	If so, animate the sprite backwards.
IF FRAME = 0	If this is the first frame,
LET PARAMB 0	set to animate forwards.
ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 9	If this is the boiler room jey,
IF A = 0	Is it time to animate?
IF PARAMB = 0	If so, is the sprite being animated forward?
ANIMATE	If so, animate the sprite.
IF FRAME = 4	If this is the last frame,
LET PARAMB 1	set to animate backwards.
ENDIF	
ELSE	Is the sprite being animated backwards?
ANIMBACK	If so, animate the sprite backwards.
IF FRAME = 0	If this is the first frame,
LET PARAMB 0	set to animate forwards.
ENDIF	
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 8	If this is a heart,
SPRITEINK 2	Red.
SUBTRACT 8 FROM X	
SUBTRACT 8 FROM Y	
SPRITEINK 66	Bright red.
ADD 8 TO X	
ADD 8 TO Y	
IF A = 0	Is it time to animate?
LET FRAME 0	Reset the current frame.
IF PARAMB = 10	Is this frame 10 of the animation sequence?
LET FRAME 1	If so, set the frame image to 1.
BEEP 2	Beep!
ENDIF	
IF PARAMB = 12	Is this frame 12 of the animation sequence?
LET FRAME 1	If so, set the frame image to 1.
BEEP 2	
ENDIF	
ADD 1 TO PARAMB	Increment the frame sequence counter.
IF PARAMB > 12	If the frame sequence counter is > 12,
LET PARAMB 0	reset the frame sequence counter to 0
ENDIF	
ENDIF	
IF COLLISION 0	If there is a collision with the player,
REMOVE	remove the sprite (i.e. the object has been collected).
ENDIF	

## Sprite type 7 (Teleporter)

IF A = 0	Is it time to animate?
GETRANDOM 2	Get a random number (0 or 1).

IF RND = 1	If the random number is 1,
SPRITEINK 5	set the sprite ink to cyan
ELSE	If the random number is 0,
SPRITEINK 69	set the sprite ink to bright cyan
ENDIF	(This causes the teleporter to “shimmer”).
IF PARAMB <= 3	The following lines of code control the animation of the
LET FRAME PARAMB	teleporter. Wherever you see this sort of thing in the code,
ELSE	it is used to same memory. Each frame of a sprite uses 256
IF PARAMB = 4	bytes. This code saves 6 frames, which equates to 1.5K of
LET FRAME 2	memory. The code itself uses a lot less than 1.5K, therefore
ENDIF	saving some vital resources.
IF PARAMB = 5	
LET FRAME 1	
ENDIF	
IF PARAMB = 6	
LET FRAME 0	
ENDIF	
IF PARAMB = 7	
LET FRAME 4	
ENDIF	
IF PARAMB = 8	
LET FRAME 5	
ENDIF	
IF PARAMB = 9	
LET FRAME 6	
ENDIF	
IF PARAMB = 10	
LET FRAME 5	
ENDIF	
IF PARAMB = 11	
LET FRAME 4	
ENDIF	
ENDIF	
ADD 1 TO PARAMB	Increment the frame sequence counter.
IF PARAMB > 11	If this is the last frame,
LET PARAMB 0	reset counter to 0.
ENDIF	
ENDIF	

### Sprite type 8 (Extraction Fan)

IF A = 0	Is it time to animate?
LET PARAMB 0	Default fan to off.
IF PARAMA = 0	Fan 1.
IF GOT 11	If got Battery 1,
LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF PARAMA = 1	Fan 2.
IF GOT 12	If got Battery 2,
LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF PARAMA = 2	Fan 3.
IF GOT 13	If got Battery 3,
LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF PARAMA = 3	Fan 4.
IF GOT 14	If got Battery 4,
LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF PARAMA = 4	Fan 5.
IF GOT 15	If got Battery 5,

LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF PARAMA = 5	Fan 6.
IF GOT 16	If got Battery 6,
LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF PARAMA = 6	Fan 7.
IF GOT 17	If got Battery 7,
LET PARAMB 1	then set fan to on.
ENDIF	
ENDIF	
IF GOT 18	If still have the Boiler OFF switch (ie. Boiler is on),
LET PARAMB 0	do not allow the fans to turn.
ENDIF	
IF PARAMB = 0	Is the fan off?
SPRITEINK 2	Make the fan Red
ELSE	Is the fan on?
SPRITEINK 4	Make the fan Green, and
ANIMATE	animate it.
ENDIF	
ENDIF	

## Initialise sprite

LET PARAMA 0	
LET PARAMB 0	
IF TYPE = 0	If the current sprite is the player,
LET G 50	then start in "God" mode.
LET F 4	
ENDIF	
IF TYPE = 1	If this is Evil Bob,
IF D <= 1	If D, the colour counter, has not already been set,
IF SCREEN <= 13	Is the player in the house?
LET D 4	First sprite should be green
ELSE	Otherwise, if the player is outside
LET D 2	First sprite should be red
ENDIF	
LET PARAMB D	Assign sprite colour.
ADD 1 TO D	Increment to the next colour
IF D > 5	If the colour counter is > 5,
LET D 2	reset the colour counter to 2 (Red)
ENDIF	
ENDIF	
IF TYPE = 3	If this is a spider,
IF D <= 1	If D, the colour counter, has not already been set,
LET D 3	start at magenta.
ENDIF	
LET PARAMB D	Assign sprite colour
ADD 1 TO D	Increment to the next colour
IF D > 5	If the colour counter is > 5,
LET D 2	reset the colour counter to 2 (Red)
ENDIF	
ENDIF	
IF TYPE = 5	If this is a collectable object,
IF IMAGE = 12	If this is the front door key sprite,
IF SCREEN = 9	and we are on screen 9,
IF GOT 3	and we have already picked up the front door key,
REMOVE	then remove the front door key sprite from the screen.
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 11	If this is the balcony key sprite,

IF SCREEN = 19	and we are on screen 19,
IF GOT 2	and we have already picked up the balcony key,
REMOVE	then remove the balcony key sprite from the screen
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 10	If this is the air con room key sprite,
IF SCREEN = 10	and we are on screen 10,
IF GOT 1	and we have already picked up the air con room key,
REMOVE	then remove the air con room key sprite from the screen.
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 9	If this is the boiler room key sprite,
IF SCREEN = 5	and we are on screen 5,
IF GOT 0	and we have already picked up the boiler room key,
REMOVE	then remove the boiler room key sprite from the screen.
ENDIF	
ENDIF	
ENDIF	
IF IMAGE = 8	If this is a heart,
IF SCREEN = 0	and we are on screen 0,
IF D <= 7	If the sprite counter <= 7,
LET D 7	set the sprite counter to 7
ENDIF	
LET PARAMA D	Number the sprite so it can be matched to an object.
IF PARAMA = 7	If this is the sprite numbered 7,
IF GOT 7	and we have already picked up this heart (object 7),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
IF PARAMA = 8	If this is the sprite numbered 8,
IF GOT 8	and we have already picked up this heart (object 8),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
ADD 1 TO D	Increment the sprite counter.
ENDIF	
IF SCREEN = 12	If this is screen 12,
IF GOT 4	and we have already picked up this heart (object 4),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
IF SCREEN = 30	If this is screen 30,
IF GOT 5	and we have already picked up this heart (object 5),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
IF SCREEN = 7	If this is screen 7,
IF GOT 6	and we have already picked up this heart (object 6),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
IF SCREEN = 18	If this is screen 18,
IF GOT 9	and we have already picked up this heart (object 9),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
IF SCREEN = 26	If this is screen 26,
IF GOT 10	and we have already picked up this heart (object 10),
REMOVE	then remove this heart sprite from the screen.
ENDIF	
ENDIF	
ENDIF	
ENDIF	

IF SCREEN = 1	If this is screen 1,
IF TYPE = 8	and if the current sprite is a fan,
LET PARAMA D	then number it so that it can be matched to a battery.
ADD 1 TO D	Increment the battery counter.
ENDIF	
ENDIF	

## Main loop 1

IF P = 99	Is the game in attract mode?
IF SCREEN <= 30	
LET LINE 23	
LET COLUMN 4	
IF K = 1	If the message display counter is 1,
COLOUR 7	
MESSAGE 40	"O=Left, P=Right, Z=Jump"
ENDIF	
IF K = 50	If the message display counter is 50,
COLOUR 5	
MESSAGE 41	"Press JUMP to start"
ENDIF	
IF K = 100	If the message display counter is 100,
COLOUR 1	
LET COLUMN 1	
MESSAGE 11	Copyright message.
ENDIF	
ADD 1 TO K	Increment the message display counter.
IF K = 151	If the message display counter = 151,
LET K 1	reset it to 1.
ENDIF	
ENDIF	
IF KEY 4	Has the JUMP key been pressed?
COLOUR 1	
LET LINE 23	
LET COLUMN 1	
MESSAGE 11	Copyright message.
LET SCREEN 0	Start the game at screen 0.
LET A 0	Reset the animation indicator.
LET L 0	Reset
LET P 1	Reset play mode indicator (1 = play).
LET X 144	Reset player Y position (yes, they are swapped in AGD).
LET Y 120	Reset player X position (yes, they are swapped in AGD).
SPAWN 0 0	Spawn the player sprite (0) with image 0.
EXIT	Exit immediately.
ENDIF	
IF E = 0	Is it time to change the screen?
ADD 1 TO C	Incr the screen counter (C).
IF C = 32	We only have 31 screens, so
LET C 2	reset the screen counter.
ENDIF	
LET E 151	Reset E for 151 cycles.
LET SCREEN C	Display the next screen in the sequence.
ENDIF	
SUBTRACT 1 FROM E	Decrement the screen timer (E).
ENDIF	
IF A = 1	Animation flip flop.
LET A 0	
ELSE	
ADD 1 TO A	
ENDIF	
IF SCREEN = 31	Is the current screen the home screen?
IF L = 0	Has the messages already been displayed? If not,
LET E 255	Set the screen change timer to 255 cycles.
LET K 0	



LET L 1	Set the message displayed indicator.
LET P 99	Set the game to attract mode.
LET LINE 4	
LET COLUMN 16	
MESSAGE 42	"Silly BOB has"
LET LINE 5	
LET COLUMN 16	
MESSAGE 43	"lost the keys"
LET LINE 6	
LET COLUMN 16	
MESSAGE 44	"to the Boiler"
LET LINE 7	
LET COLUMN 16	
MESSAGE 45	"and Air Con"
LET LINE 8	
LET COLUMN 16	
MESSAGE 46	"rooms."
LET LINE 10	
LET COLUMN 16	
MESSAGE 47	"It's summer,"
LET LINE 11	
LET COLUMN 16	
MESSAGE 48	"and it's hot."
LET LINE 13	
LET COLUMN 16	
MESSAGE 49	"Help BOB find"
LET LINE 14	
LET COLUMN 16	
MESSAGE 50	"those keys,"
LET LINE 15	
LET COLUMN 16	
MESSAGE 51	"and cool down."
COLOUR 7	
LET LINE 20	
LET COLUMN 4	
MESSAGE 40	"O=Left, P=Right, Z=Jump"
COLOUR 232	
LET LINE 23	
LET COLUMN 1	
MESSAGE 11	Copyright message.
ENDIF	
ENDIF	
IF SCREEN = 3	Is the current screen the boiler room?
IF A = 0	Is it time to animate?
IF GOT 18	Has the player got the Off switch (therefore furnace is on)?
IF L = 0	Here, L is used as a message counter. If it hasn't been set
LET L = 27	then set it to 27 (first flame in the sequence).
ENDIF	
COLOUR 66	PAPER BLACK, INK RED, BRIGHT
LET LINE 19	
LET COLUMN 24	
MESSAGE L	Flame!
ADD 1 TO L	Get the next message (flame sequence)
COLOUR 86	PAPER RED, INK YELLOW, BRIGHT
LET LINE 20	
LET COLUMN 24	
MESSAGE L	Flame!
IF L = 32	If the last flame in the sequence has been used,
LET L 27	reset the counter to the first frame (27).
ENDIF	
ELSE	If the player has the Off switch, hide the flames.
COLOUR 0	PAPER BLACK, INK BLACK
LET LINE 19	
LET COLUMN 24	
MESSAGE 33	SPACES

LET LINE 20	
LET COLUMN 24	
MESSAGE 33	SPACES
ENDIF	
ENDIF	
ENDIF	

## Main loop 2

IF 0 > 0	
IF K = 0	
LET K 4	
ENDIF	
IF J = 0	
LET J 51	Set countdown timer to 51 ticks
IF 0 = 1	God mode?
LET G 50	Set god counter to 50 ticks
LET F 4	Set flash counter to 4 ticks
LET P 0	Set player mode to 0 (can't kill)
ENDIF	
IF 0 = 2	Add life?
IF LIVES <= 7	If there are still lives to add:
ADD 1 TO LIVES	Increment the lives counter.
ENDIF	
ENDIF	
IF 0 = 7	Add battery?
IF B <= 6	If there are still batteries to add:
ADD 1 TO B	Increment the battery counter.
ENDIF	
ENDIF	
IF 0 > 2	
IF 0 <= 7	
COLOUR 7	
LET LINE 23	
LET COLUMN 1	
IF 0 = 3	
MESSAGE 38	"The BOILER ROOM key."
ENDIF	
IF 0 = 4	
MESSAGE 39	"The AIR CON ROOM key."
ENDIF	
IF 0 = 5	
MESSAGE 37	"The BALCONY key."
ENDIF	
IF 0 = 6	
MESSAGE 36	"The FRONT DOOR key."
ENDIF	
ENDIF	
ENDIF	
IF 0 = 8	Display all collected batteries in screen 1.
LET O 0	Reset O indicator so other actions can take place.
LET J 0	Reset J countdown timer as we don't need it.
LET K 0	
IF H = 0	Have these messages not already been displayed?
LET H 1	Set the message displayed indicator.
IF GOT 19	If the furnace has been turned off?
IF B <= 6	If all batteries have not been collected.
COLOUR 6	PAPER 0 (Black), INK 6 (Yellow)
LET LINE 14	
LET COLUMN 5	
MESSAGE 20	"Collect all of the"
LET LINE 15	
LET COLUMN 5	
MESSAGE 21	"batteries to cool the"

LET LINE 16	
LET COLUMN 5	
MESSAGE 22	"whole house. Look in"
LET LINE 17	
LET COLUMN 5	
MESSAGE 23	"the garden maze."
ELSE	If all batteries have been collected.
COLOUR 4	PAPER 0 (Black), INK 4 (Green)
LET LINE 14	
LET COLUMN 4	
MESSAGE 24	"Congratulations. You have"
LET LINE 15	
LET COLUMN 4	
MESSAGE 25	"cooled the whole house."
LET LINE 15	
LET COLUMN 4	
MESSAGE 26	"You may now have a break."
ENDIF	
ELSE	If the furnace is still on.
COLOUR 7	
LET LINE 14	
LET COLUMN 6	
MESSAGE 34	"You need to turn off"
LET LINE 15	
LET COLUMN 6	
MESSAGE 35	"the boiler first."
ENDIF	
COLOUR 7	
IF GOT 11	
LET LINE 10	
LET COLUMN 3	
MESSAGE 18	Top of battery.
LET LINE 11	
LET COLUMN 3	
MESSAGE 19	Bottom of battery.
ENDIF	
IF GOT 12	
LET LINE 10	
LET COLUMN 7	
MESSAGE 18	Top of battery.
LET LINE 11	
LET COLUMN 7	
MESSAGE 19	Bottom of battery.
ENDIF	
IF GOT 13	
LET LINE 10	
LET COLUMN 11	
MESSAGE 18	Top of battery.
LET LINE 11	
LET COLUMN 11	
MESSAGE 19	Bottom of battery.
ENDIF	
IF GOT 14	
LET LINE 10	
LET COLUMN 15	
MESSAGE 18	Top of battery.
LET LINE 11	
LET COLUMN 15	
MESSAGE 19	Bottom of battery.
ENDIF	
IF GOT 15	
LET LINE 10	
LET COLUMN 19	
MESSAGE 18	Top of battery.
LET LINE 11	

LET COLUMN 19	
MESSAGE 19	Bottom of battery.
ENDIF	
IF GOT 16	
LET LINE 10	
LET COLUMN 23	
MESSAGE 18	Top of battery.
LET LINE 11	
LET COLUMN 23	
MESSAGE 19	Bottom of battery.
ENDIF	
IF GOT 17	
LET LINE 10	
LET COLUMN 27	
MESSAGE 18	Top of battery.
LET LINE 11	
LET COLUMN 27	
MESSAGE 19	Bottom of battery.
ENDIF	
ENDIF	
EXIT	
ENDIF	
ENDIF	
SUBTRACT 1 FROM J	Dec countdown timer.
IF 0 = 1	
IF K > 2	
COLOUR 74	
ELSE	
COLOUR 73	
ENDIF	
IF J = 1	
COLOUR 73	
ENDIF	
ELSE	
IF K > 2	
COLOUR 79	
ELSE	
COLOUR 73	
ENDIF	
IF J = 1	
COLOUR 79	
ENDIF	
ENDIF	
SUBTRACT 1 FROM K	
LET LINE 2	
IF 0 = 1	
LET COLUMN LIVES	
MESSAGE 17	
ENDIF	
IF 0 = 2	
LET COLUMN LIVES	
MESSAGE 8	
ENDIF	
IF 0 = 3	
LET COLUMN 24	
MESSAGE 12	
ENDIF	
IF 0 = 4	
LET COLUMN 25	
MESSAGE 13	
ENDIF	
IF 0 = 5	
LET COLUMN 26	
MESSAGE 14	
ENDIF	

IF 0 = 6	
LET COLUMN 27	
MESSAGE 15	
ENDIF	
IF 0 > 6	
IF 0 <= 8	
LET COLUMN 28	
DISPLAY B	
LET COLUMN 29	
MESSAGE 16	
ENDIF	
ENDIF	
IF J = 1	Is the countdown timer at the end of its cycle?
IF 0 = 1	Remove a life?
SUBTRACT 1 FROM LIVES	
ENDIF	
IF 0 > 2	
IF 0 <= 7	
COLOUR 1	
LET LINE 23	
LET COLUMN 1	
MESSAGE 11	Copyright message.
ENDIF	
ENDIF	
LET 0 0	All done, so reset to 0.
LET K 0	
ENDIF	
ENDIF	

## Game initialisation

GET 18	Get boiler "OFF" switch from screen 3, turning boiler on.
LET LIVES 1	
COLOUR 71	
CLS	
LET A 0	
LET B 0	
LET C 1	
LET D 0	
LET L 0	
LET M 99	
LET P 1	
COLOUR 1	
LET LINE 23	
LET COLUMN 1	
MESSAGE 11	Copyright message.
COLOUR 78	
LET LINE 1	
LET COLUMN 1	
MESSAGE 1	"The Adventures of Bouncing Bob"
COLOUR 77	
LET LINE 2	
LET COLUMN 1	
MESSAGE 2	"Summer Breeze"

## Restart screen

LET D 0	
LET H 0	
LET L 0	
IF SCREEN = 0	
COLOUR 79	
LET LINE 2	

LET COLUMN 1	
MESSAGE 8	
ENDIF	
IF SCREEN = 31	
COLOUR 79	
LET LINE 2	
LET COLUMN 1	
MESSAGE 9	
ENDIF	

**Fell too far**

**Kill player**

**Lost game**

**Completed game**

### **Global Variables (A to P)**

- A. Animation indicator. Slows down sprite animation without slowing down game speed. Animate when 0, otherwise do not animate.
- B. Battery collected counter. Counter is incremented every time a battery is collected.
- C. Current screen number during attract mode.
- D. Fan (Sprite: Image 4, Type 8) counter and sprite colour counter, for use in the Initialise Sprite event.
- E. Screen change counter during attract mode.
- F. Flash indicator. Used in conjunction with G to flash the player sprite, indicating that the player is invincible.
- G. God indicator. Gives the player invincibility for a short time at the start of the game, and also after a player's life has been taken.
- H. Screen 1 (Air Con Room) message and battery displayed indicator.
- I. Temporary variable.
- J. Status Icon flash timer.
- K. Flash indicator. Used in conjunction with J to flash status icons.
- L. Door removed indicator (Screens 0, 7, and 10) / Fire counter (Screen 3), message displayed indicator (Screen 31).
- M. Message indicator. Allows the display of messages at the bottom of the screen. When 98, the message area is cleared (or default message is displayed). When 99, no action is taken. When  $M \geq 100$ , then message  $M$  minus 100 is displayed.
- N. Message displayed indicator. When 1, a message is being displayed. Used in conjunction with M. When M is 98, and N is 1, the message area is cleared (or default message is displayed).

- O. Output indicator. To output special events, such as flashing messages. 0 = do nothing, 1 = flash life before removing, 2 = flash life when adding, 3 = flash right key when collected, 4 = flash left key when collected, 5 = flash balcony key when collected, 6 = flash front door key when collected, 7 flash battery when collected, 8 to display messages and batteries in screen 1, 9 to allow the player to move screens without getting killed by sprites from the previous screen. All of these events are controlled through the 2<sup>nd</sup> main event loop.
- P. Player indicator. When 0, player sprite is dying, otherwise player sprite is alive. This always the display of a separate sprite animation when player has been killed.

## Miscellaneous

**Collision distance:** 12

**Sprite height:** 16

**SPRITEINK mask:** 71

**Order sprites:** ON

**Specialise:** PARTICLES

**AY sounds:** OFF

## Objects

See the “Objects” folder for images.

- |                                      |                        |
|--------------------------------------|------------------------|
| 0 - Key for the right basement door. | 10 - Heart 7           |
| 1 - Key for the left basement door.  | 11 - Battery 1         |
| 2 - Key for the balcony door.        | 12 - Battery 2         |
| 3 - Key for the front door.          | 13 - Battery 3         |
| 4 - Heart 1                          | 14 - Battery 4         |
| 5 - Heart 2                          | 15 - Battery 5         |
| 6 - Heart 3                          | 16 - Battery 6         |
| 7 - Heart 4                          | 17 - Battery 7         |
| 8 - Heart 5                          | 18 - Boiler on switch  |
| 9 - Heart 6                          | 19 - Boiler off switch |

## Sprites

See the “Sprites” folder for images.

- 0 - Bouncing Bob - The player.
- 1 - Bat.
- 2 - Spider.
- 3 - Dying Bob.

- 4 - Extractor fan.
- 5 - Teleport.
- 6 - "Frog" enemy - platform horizontal - moving right.
- 7 - "Frog" enemy - platform horizontal - move left.
- 8 - Beating heart.
- 9 - Rotating key - right basement door.
- 10 - Rotating Key - left basement door.
- 11 - Rotating key - balcony door.
- 12 - Rotating key - front door.
- 13 - Evil bob.



## Text

0	GAME OVER
1	The Adventures of Bouncing Bob
2	Summer Breeze ”
3	“
4	Find the Boiler Room key.
5	Find the Air Con Room key.
6	Find the Balcony key.””
7	Find the Front Door key.
8	!
9	“
10	“””
11	“ #2015 retrific.com “
12	\$
13	%
14	&
15	‘
16	(
17	)
18	*+
19	, -
20	Collect all of the
21	batteries to cool the
22	whole house. Look in
23	the garden maze.
24	Congratulations. You have
25	cooled the whole house.

26	You may have a break
27	:<>;=
28	<>:=?
29	>:<?;
30	;=?:<
31	=?;<>
32	?;=>:
33	“””””
34	You need to turn off
35	the boiler first.
36	The FRONT DOOR key. “
37	The BALCONY key. “
38	The BOILER ROOM key. “
39	The AIR CON ROOM key. “
40	O]Left[ P]Right[ Z]Jump
41	“””Press JUMP to start””
42	Silly BOB has
43	lost the keys
44	to the Boiler
45	and Air Con
46	rooms.
47	It@s summer[
48	and it@s hot.
49	Help BOB find
50	those keys[
51	and cool down.

Some characters used in the test, above, have been redefined in the character set:

!	Smiley face
“	Space
#	Copyright symbol
\$	Right key
%	Left key
&	Balcony key
‘	Front door key
(	x battery

)	Surprised Bob
*+, -	Battery
/	Not used!
:<>;=?	Flames
@	Apostrophe (‘)
[	Comma (,)
]	Equals sign (=)