Operating System Commands

Key pressed	Result
esc	Start running demo if focus is on a demo box
F1	Start execution, or do a step of execution
F2	Execute until the next breakpoint is reached
F3	Toggle the breakpoint status of the focus box
F4	Halt execution
F5	Save the current universe to the file
F6	Save the current universe to the file and exit the program (or interrupt demo if a demo is running)
F7	Switch between normal viewing and a single viewer aimed at the console box
F8	Switch between having one interactive viewer and having one smaller interactive viewer, a viewer aimed at the console box, and a viewer aimed at the ports box
F9	Make the interactive viewer show a larger area of box reality
F10	Make the interactive viewer show a smaller area of box reality
PgUp	Move the interactive viewer up
PgDn	Move the interactive viewer down
Home	Move the interactive viewer to the left
End	Move the interactive viewer to the right

Mouse action	Result
Drag mouse	Move the interactive viewer with the mouse

Navigation

Key pressed	Result
ctrl+i (or ↓)	Move the focus **i**n
ctrl+o (or ↑)	Move the focus **o**ut
ctrl+n (or →)	Move the focus to the **n**ext box in the inner list
ctrl+p (or ←)	Move the focus to the **p**revious box in the inner list
ctrl+c (or del)	**C**hange the focus box to show a different aspect
ctrl+j (or ins)	**J**ump to a different part in the focus box
ctrl+f	Move focus to **f**irst symbol in current string part
ctrl+l	Move focus to **l**ast symbol in current string part
ctrl+u	Move focus **u**p a line—documentation part only
ctrl+d	Move focus **d**own a line—documentation part only

Mouse action	Result
Left-click mouse	Move focus to box or symbol clicked
Right-click mouse	Move focus to box clicked and change aspect

Destruction

Key pressed	Result
ctrl+k (or backspace)	"kill" the focus box, if it is allowed to be
ctrl+w	"wipe" the focus box, which leaves structure intact while removing full aspect. Use to remove one of a pair in branch box

Creation Using Java Box Approach

Key(s)	Context	Result
any printable key	name part of java box, documentation string, string literal, char literal	the symbol is inserted in the string
any letter, digit, underscore	name string of a box	the symbol is inserted in the string
any letter, digit, underscore, [	type string of a box	the symbol is inserted in the string
space	inside container box	insert a new class box or data box or method box
space	inside sequence box	insert a new empty box
space	inside branch box, on lower row	insert a pair of empty boxes
j	empty box	convert the empty box to a java box
j	inside sequence box	insert a java box in the sequence box
b	inside sequence box	insert a branch box in the sequence box
w	inside sequence box	insert a while loop box in the sequence box
d	inside sequence box	insert a do loop box in the sequence box
f	inside sequence box	insert a for loop box in the sequence box
ctrl x	most user-creatable boxes	"xerox" the focus box to clipboard
ctrl g	anywhere copied box makes sense	"glue" the clipboard box on the empty box or insert copy of box

Miscellaneous Interactive Operations

Key(s)	Context	Result
ctrl+f, ctrl+l	name part of java box, documentation string, value string in string type data box	move cursor to first, last symbol in string
ctrl+u, ctrl+d	name part of java box, documentation string	move cursor up, down a row
ctrl+t	java box or inside a java box	manually translate the java box
ctrl+r	any box	"remarkize" from the focus inward, changing aspects to documentation
ctrl+s	any box	"show" from the focus inward, changing aspects to full

Binary Operations

Operation string	Name	Description
+	addition	adds two numbers or concatenates two strings
-	subtraction	subtracts second number from first number
*	multiplication	multiplies two numbers
/	division	divides two `float` values, does whole number division of two `int` values, second value cannot be 0
%	remainder	values must be `int`, produces remainder when divide first by second
<	less than	compares two numbers, two `char` values, or two `string` values, produces `boolean`
>	greater than	(similar to < but using >)
<=	less than or equal to	(similar to < but using $\leq$)
>=	greater than or equal to	(similar to < but using $\geq$)
==	equal to	(similar to < but using =)
!=	not equal	(similar to < but using $\neq$)
&	logical and	produces `true` if both are `true`, `false` otherwise
|	logical or	produces `false` if both are `false`, `true` otherwise

Unary Operations

Operation string	Name	Description
-	opposite	switches the sign of the number
!	not	switches the boolean value between true and false

Primitive Data Types

Type name	Full name	Sample Values
int	integer	-37, 28, 0, 123456789
float	floating point number	3.14, -12.73e-4
char	character (symbol)	a, ?, B, %
boolean	boolean	false, true
string	string	hello, 3.14, B, -37

System Methods

Name	Number of arguments	Action
Screen methods		
`_print`	≥ 1	Displays all arguments in the console box, starting at the cursor location
`_clear`	0	Clears the console box and puts the cursor at upper-left corner
`_moveTo`	2	Moves the cursor to the row and column specified by the first and second arguments, respectively
`_whatRow`	0	Returns the row in which the cursor is located
`_whatCol`	0	Returns the column in which the cursor is located
`_hideCursor`	0	Makes the cursor not show up
`_showCursor`	0	Makes the cursor show up
Input methods		
`_get`	0	Waits for user to press a key, that `char` becomes its value
`_getLine`	0	Waits for user to type a `string`, terminated by the enter key, returns that `string` as its value

Conversion methods

`_char`	1	Converts the argument, which must be an `int` between 0 and 255, to the corresponding `char` value
`_ascii`	1	Returns the `int` ASCII code for the argument, which must be a `char`
`_isInt`	1	Returns true or false depending on whether the `string` argument represents a legal `int` value
`_isFloat`	1	Returns true or false depending on whether the `string` argument represents a legal `float` value
`_toInt`	1	Returns the `int` value obtained by converting the `string` argument to an `int` (error if it is not a legal integer)
`_toFloat`	1	Returns the `float` value obtained by converting the `string` argument to a `float` (error if it is not a legal real number value)
`_intToString`	1	Returns the `string` obtained by converting the `int` argument to a `string`
`_floatToString`	1	Returns the `string` obtained by converting the `float` argument to a `string`
`_charToString`	1	Returns the `string` obtained by converting the `char` argument to a `string`
`_charsToString`	2	The first argument must be a `char` array, and the second argument must be a number between 0 and the length of the array, returns a `string` built by copying specified number of `char`s from the array

Mathematical methods

`_abs`	1	Returns the absolute value of the `float` argument
`_sqrt`	1	Returns the square root of the `float` argument (error if argument is negative)
`_sin`	1	Returns the sine of the `float` argument which is an angle measured in radians
`_cos`	1	Returns the cosine of the `float` argument
`_tan`	1	Returns the tangent of the `float` argument (error if argument is odd multiple of $\pi/2$)

`_asin`	1	Returns an angle in radians whose sine is the argument, argument must be in $[-1, 1]$
`_acos`	1	Returns an angle whose cosine is the argument (error if argument is not in $[-1, 1]$
`_atan`	1	Returns an angle whose tangent is the argument
`_exp`	1	Returns e^x where x is the argument
`_log`	1	Return the natural logarithm of the argument (error if argument is not positive)
`_floor`	1	Returns the largest `int` that is less than or equal to the `float` argument
`_ceil`	1	Returns the smallest `int` that is greater than or equal to the `float` argument
`_random`	0	Produces a psuedo-random `int` value between 1 and 32767
`_seed`	1	Starts the psuedo-random sequence at the argument which must be an `int` between 1 and 32767

File methods

`_open`	3	The first argument must be an `int` port number that is between 1 and 4, the second argument must be a `string` that is either "input" or "output" and the third argument must be a `string` that is the name of the file that is being opened
`_close`	1	Close the file that is connected to the port whose number is the `int` argument
`_fget`	1	Return the `char` that is read from the file connected to the port whose number is the `int` argument
`_fput`	2	The first argument is the `int` port number and the second argument is the `char` that is written to the file connected to the port specified by the first argument
`_eof`	1	The argument is a port number that must be connected to a file open for input, and a `boolean` value is returned depending on whether the file has been entirely read

Reference methods

`_null`	0	Produces the special non-existent reference #0
`_this`	0	Returns the reference of the instance box in the heap box whose method is executing
`_destroy`	1	The first argument is a reference to an instance box in the heap box. That instance box is destroyed.

Miscellaneous methods

`_length`	1	Returns the number of symbols in the argument `string`
`_charAt`	2	The first argument is a `string` and the second is an `int` that must be between 0 and one less than the length of the `string`. Returns the `char` in the `string` at the specified position
`_progChar`	2	The first argument is an `int` between 133 and 255, which is the ASCII code of the `char` that is having its image programmed. The second argument is a `string` that tells how to draw the new image. The string can have a sequence of zero or more commands of the form `Txyuv` where `T` is replaced by `L` to mean "draw a line from (x, y) to (u, v)," or `R` to mean "draw a rectangle with corners at (x, y) and (u, v)," or `F` to mean "draw a filled rectangle with corners at (x, y) and (u, v)," where x and u are x coordinates ranging from a to q and y and v are y coordinates ranging from a to s
`_halt`	0	Halt execution