credits, disclaimer, license

"L in Cubeland" has been programmed by poulos poulopoulos. the day-time and night-time music themes are "passepied" and "clair de lune", respectively, from claude debussy's "suite bergamasque", orchestrated by isao tomita, as they appeared in the disk "snowflakes are dancing" (1974). the finished game music theme is "neptune, the mystic" from gustav holst's "the planets", orchestrated by isao tomita, as it appeared in the disk "the planets" (1976).

the publisher gives no warranty about the facts mentioned in the following sections nor about facts that are not mentioned.

you can do whatever you want with this piece of software and its parts, provided that it does not lead to:

- commercial use of them or
- hiding/altering their creators' identities

in particular, a copy of this document must be included in every relevant distribution.

game description

L, the player, lives in Cubeland, a cubic gridlike space. each grid-position of Cubeland can contain one cube or be empty. in a normal state, L stands on a cube aligned with its edges and the 2 (its height) grid-positions that it occupies are empty. it can turn left/right and step back/front. each turn is of 90 degrees. each step can be a horizontal move of 1 position (walking), an upward move of 1 position followed by a horizontal move of 1 position (climbing up) or a horizontal move of 1 position followed by a downward move of 1 position (climbing down), depending on the cubes around L. in a situation that L does not stand on a cube, it falls down or, when it has reached the bottom of Cubeland, gets positioned on the lowest cube in the central grid-column that has enough empty space above it.

L can destroy a cube right in front of it, up to its height, or push it forward, possibly moving some other cubes too. it can also destroy all cubes in the grid-column in front of it. moreover it can make all the cubes that have empty space below them collapse (fall down), until there is no such cube. cubes that reach the bottom of Cubeland or hit L during a collapse disappear.

L has a bag containing cubes and lamps that it can place in Cubeland. a cube can be placed right in front of L, up to its height. a lamp can be placed in the low position in front of L. lamps can overlap with other objects in Cubeland (cubes, lamps, L). they disappear when L has made a certain number of steps after placing them in Cubeland. the maximum number of lamps that can exist in Cubeland at the same time is 5 (but there is no bound for the lamps in L's bag).

each cube placed in Cubeland has a stock of cubes, ranging from 0 to 4, from which L can take when it stands on it. a cube's stock is indicated by its color (the darker the color, the larger the stock). to obtain a lamp, L must destroy a column that contains cubes above L.

the game has a score, starting from 0 and changing according to L's actions. each level in a fall subtracts 10 points. reaching the bottom of Cubeland and getting hit by a cube during

a collapse both subtract 100 points. obtaining a cube subtracts 10 points and placing a new cube in Cubeland adds 10 points. when a column gets destroyed, each cube above L destroyed adds 10 points. in a collapse, each cube adds 1 point for each level that it falls.

the game also has stages. it starts from stage 0 and moves to the next stage every time the score exceeds a new multiple of 1000. in each new stage a sun appears above Cubeland. with every step of L, it moves by one position above one of the diagonals of Cubeland's top. it disappears when it reaches the opposite, with respect to Cubeland's center, position of its initial one. additionally, in each new stage the probability distribution that the stock of a new cube placed in Cubeland follows changes. starting from the uniform distribution, the probability of 0 stock gradually goes to 1 (never reaching it theoretically) while the probabilities of non-0 stocks go to 0, the larger the stock, the faster the pace.

in stage 0, the bottom of Cubeland gets covered by cubes with all the other positions left empty, L gets placed on the central cube, which has 0 stock and cannot be destroyed, given an empty bag, and 100 points are added to the score, leading to the next stage where L starts playing. the game ends if there are no lamps placed in Cubeland and no sun above it or the score is negative or L has reached the bottom of Cubeland and there is not enough empty space in the center for it to get positioned.

related user's controls (with keyboard language set to english):

left-right	turn left-right
down-up	step back-front
space	obtain cube
enter (no shift)	place cube in Cubeland
shift+enter	place lamp in Cubeland
x	destroy column
d	destroy cube
c	make cubes collapse
f	push cube forward

states, parameters, settings

in Cubeland there are 5 cameras, 4 on the corners of the top facing towards the center and 1 in the high position of L facing where it faces. any moment one of the top cameras is selected and either this one or L's camera is used to view the scene. the user can adjust the zoom level and the tilt of the camera in use. additionally, either all the separating grid-lines/planes or only those around L can be made visible, with the latter being full or cropped according to L's position. all the information about the cameras, grid-lines/planes visibility, L, cubes, lamps, sun, score and stage any moment that no move is taking place is called a state of the game. a state can be stored and loaded by the user, using either a file named state located in the program's storage folder (LiC\storage) or one with a custom name and location, with the same folder as a base when a relative path is given. a state

can also be loaded from the command line at the program's start, typing text of the form s=<path> after the executable file's name.

there are 4 parameters, any valuation of which defines a variation of the game. all have some default values which can be changed from the command line, typing text of the form <parameter>=<value> or loading a stored state when executing the program. the first parameter is named m and takes positive integers as values. defining $n=2\cdot m+1$, the following facts hold:

- Cubeland's size is n in each dimension
- a sun goes through $5\cdot n$ different positions (at most)
- a lamp lasts for $4\cdot m$ steps

m defaults to 5. the 3 remaining parameters, named a, d and r, take non-negative (positive for a) reals as values. they determine the probability distribution that the stock of a new cube placed in Cubeland follows in each stage. defining for every non-negative integer i $w(i)=\max(a-d\cdot\sum_{j<i} r^j,0)$, in stage s, the probability that a new cube's stock is of size i is:

$$\frac{w(i)^s}{\sum_{j<5} w(j)^s}$$

a and r default to 1.0 and d defaults to 0.15. notice that, while the comments made in the previous section about the behavior of the probability distributions are valid for these values, this behavior can be altered using different values. for example, setting d to 0.0 results in having the uniform distribution in every stage. when loading a state after the program starts, this must come from the same variation of the game as the current one.

finally, there is a collection of settings through which the user can configure the program. the whole configuration is stored by the program before quiting and loaded after starting, using a file named configuration located in the program's storage folder. the first two settings are set from the command line in the form <setting>=<value>. they are named f and w and determine the rendering method used in full-screen and window mode, respectively. their value can be one of the following:

- s, standing for single-buffered
- d, standing for double-buffered without vsync
- a, standing for double-buffered with adaptive vsync
- v, standing for double-buffered with full vsync

both default to v. all the remaining settings are set while the program is running. two sound components can be present: background music and sound effects. the user can adjust the volume for both and toggle them on/off independently. the mouse pointer can be set active, meaning that its motion is used to adjust the tilt of the camera in use, or inactive and the pointer's motion tolerance can be adjusted. in order to do the above things and to adjust the camera in use from the keyboard, a setting concerning the current adjustment must be set appropriately. information about L's bag (number of cubes and lamps) as well as basic game information (stage, score and number of remaining steps without the light disappearing) can be shown or hidden independently. as mentioned before, the program can run in full-screen or window mode. the window's position and shape can be changed in the usual manner but it cannot be "maximized".

<u>related user's controls:</u>

b-m (no shift)	store-load state with default name
shift+b-m	enter state storing-loading mode
n	start new game

q/escape	quit program
f1-2-3-4-5-6	set adjustment to music-effects-pointer-zoom-horizontal tilt-vertical tilt
f7/left click	cycle through top cameras or use selected one, if L's camera is used
f8/right click	use L's camera
f9 (no shift)	cycle through grid-lines visibility options
shift+f9	cycle through grid-planes visibility options
f11	enter full-screen/window mode
f12 (no shift)	show/hide bag information
shift+f12	show/hide basic information
home	give adjustment its default value
page up-down	increase-decrease adjustment
end	toggle adjustment on/off (only for music, effects, pointer)
middle click	activate/deactivate pointer
mouse wheel	adjust zoom for camera in use
mouse motion	with pointer activated, adjust tilt for camera in use

user's controls (overriding previous ones) in state storing/loading mode:

printable character	insert character in file path
delete-backspace	delete character at current-previous position
left-right	move cursor left-right
down-up	move cursor to start-end
enter (no shift)	store/load state and exit state storing/loading mode
shift+enter	exit state storing/loading mode

installation

the program can be used with microsoft's "windows" operating system, versions "xp" and above. an additional installer-program is provided, which apart from placing the main software components in a selected location can also install any of the following items:

- a shortcut for the main program in the "desktop" folder
- a shortcut folder in the "start menu"
- the association of some file extension with the program, for opening state-files
- a copy of the installer in the program's folder
- an uninstaller-program