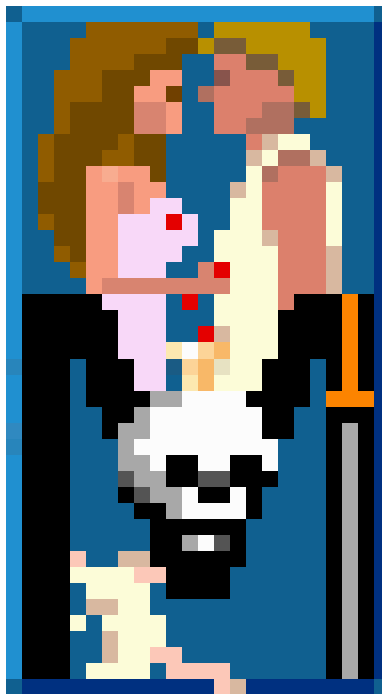


MININIM 0.9.2

The Advanced Prince of Persia Engine (a childhood dream)



Bruno Félix Rezende Ribeiro <oitofelix@gnu.org>

MININIM is the Advanced Prince of Persia Engine — a childhood dream, the free software engine for Jordan Mechner’s masterpiece, developed from scratch by Bruno Félix Rezende Ribeiro (oitofelix).

This manual is for MININIM version 0.9.2 (February 17, 2016).

Copyright © 2016 Bruno Félix Rezende Ribeiro

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License.”.

Table of Contents

Preface	1
1 Constructions	2
1.1 Non-directed relative positioning	3
1.2 Foregrounds	3
1.2.1 Floors	4
1.2.2 Pillars	6
1.2.3 Wall	7
1.2.4 Doors	7
1.2.5 Chopper	7
1.2.6 Mirror	8
1.2.7 Carpets	8
1.2.8 Arches	10
1.3 Backgrounds	11
1.4 Items	12
2 Movements	14
2.1 Directed relative positioning	15
2.2 Hangable place	15
2.3 Normal	16
2.3.1 Turn	16
2.3.2 Vertical jump	16
2.3.3 Normal jump	18
2.3.4 Walk	18
2.3.5 Descend	19
2.4 Crouch	20
2.4.1 Crouching jump	20
2.5 Run	20
2.5.1 Running jump	21
2.5.2 Running crouch	22
2.5.3 Running turn	22
2.6 Hang	22
2.6.1 Hanging fall	22
2.6.2 Hanging turn	23
2.6.3 Climb	24
2.7 Fall	24
2.7.1 Falling hang	24
2.7.2 Falling turn	25
2.8 Fight	25
2.8.1 Fight walk forward	25
2.8.2 Fight walk backward	25
2.8.3 Defense	25

2.8.4	Attack	25
2.8.5	Counter attack and counter defense	26
2.8.6	Keep sword	26
3	Key bindings	27
3.1	Game flow	27
3.2	Graphical rendering	27
3.3	Life and death	27
3.4	Kid special	28
3.5	Time	28
3.6	Skills	28
3.7	Level navigation	28
3.8	Files	29
3.9	Peripherals	29
3.10	Miscellaneous	29
4	Configuration	30
4.1	Configuration options	31
4.2	Level options	31
4.3	Time options	31
4.4	Skills options	31
4.5	Rendering options	32
4.6	Window options	32
4.7	Paths options	33
4.8	Other options	33
4.9	Help options	33
5	Software package	34
5.1	Description	34
5.2	Distribution	34
5.3	Getting a copy	34
5.4	Contact	35
5.5	Bug reporting	35
5.6	Contributing	35
5.7	Donating	35
5.8	Hacking	35
5.9	MinGW	36
6	Release	37
6.1	Announcement	37
6.2	News	38
6.3	To do	38
6.4	Thanks	38
6.5	Donors	39
Appendix A	GNU FDL	40

Appendix B Linux and the GNU system 48
Index 51

Preface

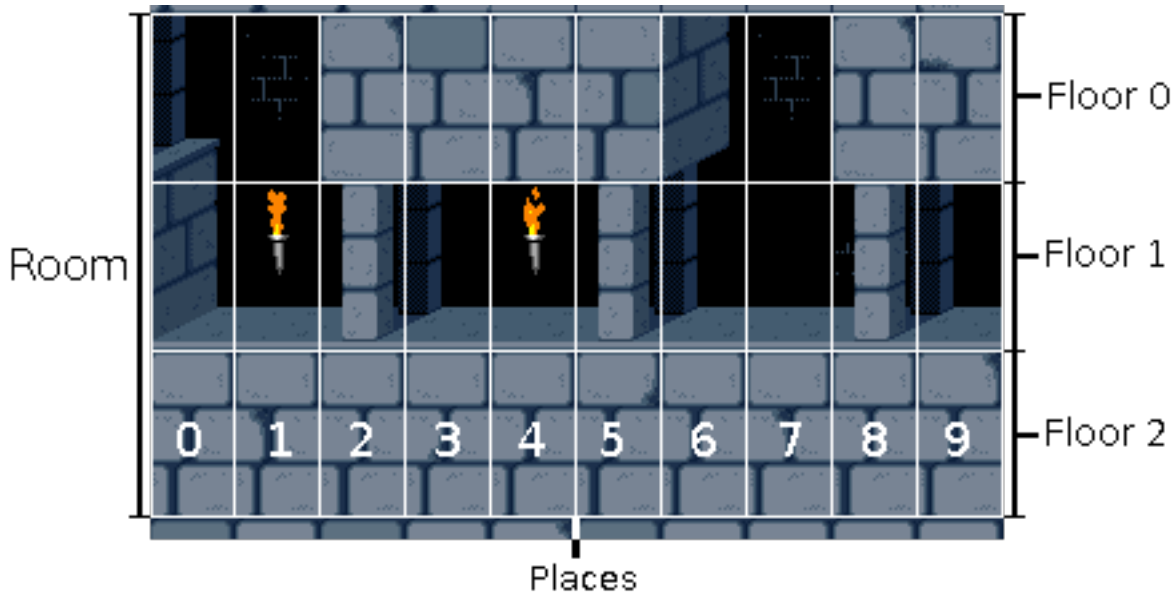
Menininho is the Brazilian Portuguese word for *male kid*, and *mininim* is how it's informally pronounced throughout the country. *Kid* is the term used by *Jordan Mechner*¹ when referring to the main character of his most famous game — **Prince of Persia** — in its source code and documentation, because the character's movements were derived from rotoscoping footage of Mechner's little brother. **MININIM** is the *Advanced Prince of Persia Engine*, written from scratch by computer programmer and free software activist *Bruno Félix Rezende Ribeiro* (`oitofelix`). Bruno had a childhood dream of deciphering the secrets of his favorite game that introduced him to computing — a dream dreamt when he was just a kid, a dream realized now that he's still a *mininim* deep in his heart.²

¹ Jordan Mechner (born June 4, 1964) is an American video game designer, author, screenwriter, and filmmaker, best known for creating the Prince of Persia video game franchise and the 8-bit computer game *Karateka*.

² Of course, none of this is reason enough to name a game engine “MININIM”, if not by the fact that it's a really cool palindrome!

1 Constructions

Each *level* is comprised of *rooms*¹. Each room is comprised of three *floors*. Each floor is comprised of 10 *places*. Each place holds a *construction* (or *con* for short).² The following figure illustrates this arrangement.



Therefore, any given construction is uniquely identified within a level by the coordinates of its place³, which are defined as the triplet *room*, *floor* and *place*: (r, f, p) .

Constructions are comprised of three elements: *foreground*, *background* and *extension*. A *construction foreground* is the physical part the kid interacts with. Physical properties determined by it are collision, hang and fall conditions, to name a few. On the other hand, a *construction background* is the part that yet visible, doesn't affect the kid in any way — its sole purpose is ornamental; torches and windows are examples of this. A *construction extension* is a modifier which tells how the con looks or behaves. For example, items and door opening states are both expressed by them. Any given construction can have only one extension. The valid extensions are:

- Item* This is a code indicating which item should be put on the construction. Used by FLOOR.
- Can't fall* This is a boolean indicating whether the construction should ignore fall conditions. Used by LOOSE_FLOOR.
- Event* This is an integer indicating which door event the construction triggers. Used by OPENER_FLOOR and CLOSER_FLOOR.

¹ The terminology used for *rooms* in the original game is *screens*.

² The terminology used for *constructions* in the original game is *tiles*.

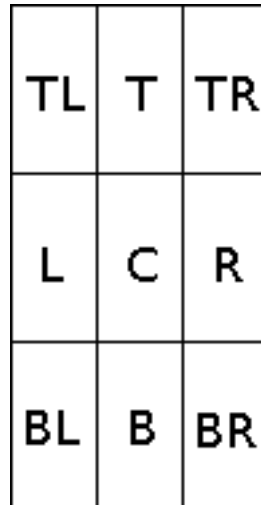
³ The term *place* is used throughout the text to describe both the last coordinate of the triplet and the world space the triplet refers to. Also, often the term *place* is omitted and the text refers just to the construction itself, implying a reference to its place, as in “the left construction”, meaning “the construction one place to the left”. Hopefully, these shouldn't be a source of ambiguity.

Step This is a integer indicating in which state, of a progressive set of configurations, a construction is in. Used by dynamic constructions that don't use other extension.

Design This is a code indicating which alternative graphical design should be used to render the construction. Used by carpet constructions.

1.1 Non-directed relative positioning

In order to easily talk about the relations between constructions it's necessary to define the *non-directed relative positioning* of a given place to its adjacent ones. Consider the place C , which has the coordinates (r, f, p) , and its 8 adjacent places, as shown in the following figure.



TL (top left) T (top) TR (top right) L (left) C (center)
R (right) BL (bottom left), B (bottom) BR (bottom right)

Then we define the following mapping between names and coordinates:

Name	Coordinates
TL	$(r, f - 1, p - 1)$
T	$(r, f - 1, p)$
TR	$(r, f - 1, p + 1)$
L	$(r, f, p - 1)$
C	(r, f, p)
R	$(r, f, p + 1)$
BL	$(r, f + 1, p - 1)$
B	$(r, f + 1, p)$
BR	$(r, f + 1, p + 1)$

1.2 Foregrounds

Construction foregrounds have physical properties that determine how they behave in certain events and also how the kid is affected by the interaction with them.

Strictly traversable

This property means that the construction can't support the kid vertically at its bottom, thus he falls immediately when trying to rest on it. Notice that cons having this property can't be *hangable*.

Traversable

This property means that the construction can't *indefinitely* support the kid vertically at its bottom, thus he *eventually* falls when trying to rest on it. Notice that *strictly traversable* is a particular case of this.

Collidable This property means that the construction prevents the kid from passing through it horizontally, thus he collides (often recoiling from the impact) when trying to do so.

Semi-collidable

This property means that the construction is *collidable* only under certain circumstances.

Hangable This property means that the kid can hang on the construction's bottom, in case there is an *strictly traversable* one at its side.

Depressible

This property means that the construction is sensible to pressure at its top, thus some reaction is triggered when the kid exerts such pressure.

Trap This property means that the construction might harm the kid.



Rigid This property means that the construction can't be smashed by a falling one, thus that which falls stops at the rigid con's top.

Dynamic This property means that the construction has a set of distinct states in which it can be in. Such states matter for how the construction looks and behaves.

It's possible that some of these properties may apply only under certain conditions of the construction and/or the kid.

1.2.1 Floors

Floors make the ground above constructions the kid stands on. All of them are *hangable*, except for NO_FLOOR.

Code	Dungeon	Palace	Notes
NO_FLOOR ⁴			This is an empty space. This is <i>strictly traversable</i> and uses no extension.
FLOOR			This is a normal floor. This uses the <i>item</i> extension.

⁴ The *no floor* con is called *empty* in the original game.

BROKEN_FLOOR⁵

As soon as a falling LOOSE_FLOOR encounters a *rigid* construction, it disappears replacing the rigid con's top by a BROKEN_FLOOR. In contrast, if it encounters a *non-rigid* and *non-traversable* construction, it disappears replacing that very con by a BROKEN_FLOOR. Finally, if it encounters a *non-rigid* but *traversable* construction, it pass through, breaking over the con in case that is not *strictly traversable*. This uses no extension.

SKELETON_FLOOR



This behaves like FLOOR but uses no extension.

LOOSE_FLOOR⁷

This is rendered as FLOOR, but shakes as the kid hits the ground, by falling (from above), or vertically jumping (from below). If the kid hits near enough it's released and falls. It might harm the kid when falling over him. This is *traversable*, *depressible*, a *trap*, *dynamic* and uses the *can't fall* extension.

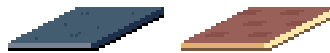
SPIKES_FLOOR



Spikes come out of it, killing the kid if he falls or steps too fast over it. This is a *trap*, *dynamic* and uses the *step* extension (ranging from 0 to 9). This gets activated as soon as the kid is in the same room and place column, and in the same or higher floor without a wall in between. Guards can activate this as well.

OPENER_FLOOR⁸

This opens one or more doors. This is *depressible*, *dynamic* and uses the *event* extension.

CLOSER_FLOOR⁹

This closes one or more doors. This is *depressible*, *dynamic* and uses the *event* extension.

⁵ The *broken floor* is called *debris* in the original game.

⁶ The original game has no palace version of the skeleton floor.

⁷ The *loose floor* is called *loose board* in the original game.

⁸ The *opener floor* is called *raise button* in the original game.

⁹ The *closer floor* is called *drop button* in the original game.

STUCK_FLOOR¹⁰

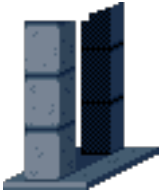

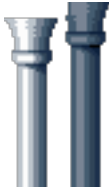

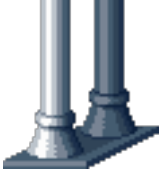
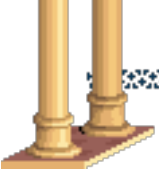


This is rendered like a pressed CLOSER_FLOOR, but it can't be unpressed. This uses no extension.

HIDDEN_FLOOR¹¹

This is rendered as NO_FLOOR, but as soon as the kid presses it, it's replaced by FLOOR. This is *depressible* and uses no extension.

1.2.2 Pillars

Pillars are aesthetic supports for the constructions on the floor above them. All of them are *hangable*, except for BIG_PILLAR_TOP. Pillars use no extension.

Code	Dungeon	Palace	Notes
PILLAR			This is <i>rigid</i> .
BIG_PILLAR_TOP			This should be placed above a BIG_PILLAR_BOTTOM. This is <i>strictly traversable</i> and <i>rigid</i> .
BIG_PILLAR_BOTTOM			This should be placed below a BIG_PILLAR_TOP.
ARCH_BOTTOM			This should be placed below an ARCH_TOP_MID.

¹⁰ The *stuck floor* is called *stuck button* in the original game.

¹¹ The original game doesn't have a HIDDEN_FLOOR construction by itself.

1.2.3 Wall

The *wall* is a solid constructs that separates sections of the level. It's *collidable*, *rigid* and uses no extension.

Code

Dungeon

Palace

WALL



1.2.4 Doors

Doors are constructions that conditionally separate levels and in-level sections. Both door types are *hangable*, *dynamic* and use the *step* extension. The extension determines the door opening level, ranging from fully open (0) to fully closed.

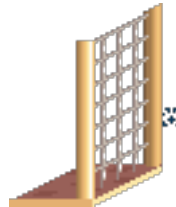
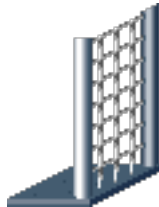
Code

Dungeon

Palace

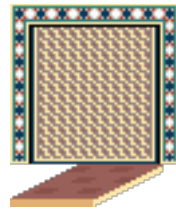
Notes

DOOR¹²



This is *semi-collidable* at right, as the kid might pass or not through it depending on the difference of the grid tip height and kid's movement height. This is *rigid* and its *step* extension range from 0 to 47.

LEVEL_DOOR¹³



Often the kid progress through levels entering these. This can only be entered when fully open. Its *step* extension range from 0 to 43.

1.2.5 Chopper

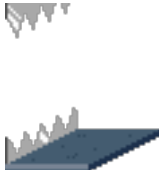
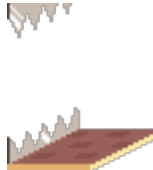
The *chopper*¹⁴ can leave the kid in half. It's a *trap*, *hangable at right*, *rigid* and *dynamic*. It uses the *step* extension which range from 0 to 4 and 128 to 132 (bloody counterpart).

¹² The *door* is called *gate* in the original game.

¹³ The *level door* is called simply *exit* in the original game. There it's divided into two constructions: *exit left* and *exit right*.



¹⁴ The *chopper* is called *chomper* in the original game.

It gets activated as soon as the kid is in the same room and floor and there is no wall in between. It doesn't get activated by guards, but can kill them as well. The chopper nullifies the *hangable at left* property of its top construction.

Code	Dungeon	Palace
CHOPPER		

1.2.6 Mirror

The *mirror* reflects the kid's image and prevents him from passing through, except if he is able to perform a running jump, whatever the side. It's *semi-collidable* at left, *hangable at right*, *rigid* and uses no extension. The mirror nullifies the *hangable at left* property of its top construction.

Code	Dungeon	Palace
MIRROR		

1.2.7 Carpets

*Carpets*¹⁵ are ornamental constructions often placed over DOOR cons. They come in two flavors: CARPET and TCARPET; the watershed difference being the *traversable* property, which the latter has but the former lacks. Both are *collidable* at right, *rigid* and make use of the *design* extension. The CARPET construction is also *hangable at left*. Also, carpets nullify the *hangable at left* property of their *right* and *top right* constructions.

Code	Dungeon	Palace	Notes
------	---------	--------	-------

¹⁵ *Carpets* are called *tapestry* in the original game.

TCARPET



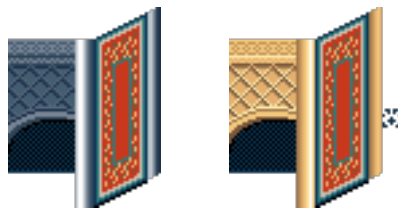
These graphics are drawn for the *design* CARPET_00.

TCARPET



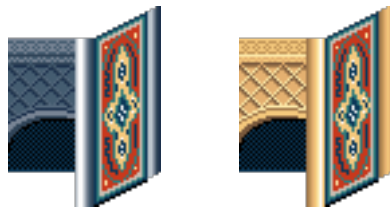
These graphics are drawn for the *design* CARPET_01.

TCARPET



These graphics are drawn for the *design* ARCH_CARPET_RIGHT_00. This is intended to end an arch. This should be placed at right of an ARCH_TOP_MID.

TCARPET



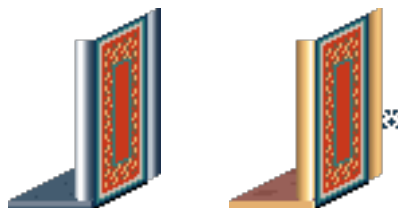
These graphics are drawn for the *design* ARCH_CARPET_RIGHT_01. This is intended to end an arch. This should be placed at right of an ARCH_TOP_MID.

TCARPET



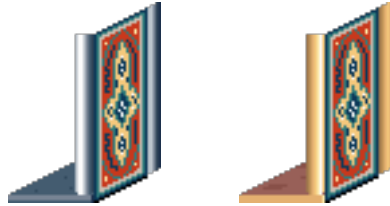
These graphics are drawn for the *design* ARCH_CARPET_LEFT. This is intended to end an arch. This should be placed two places to the left of an ARCH_TOP_MID.

CARPET



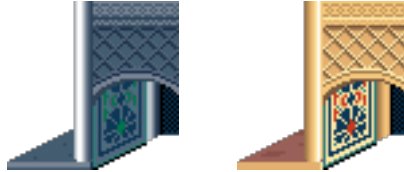
These graphics are drawn for the *design* CARPET_00.

CARPET



These graphics are drawn for the *design* CARPET_01.







CARPET



These graphics are drawn for the design ARCH_CARPET_LEFT. This is intended to end an arch. This should be placed two places to the left of an ARCH_TOP_MID.

1.2.8 Arches

*Arches*¹⁶ are ornamental constructions that can be indefinitely chained together to make a structure that spans several rooms across. They are *strictly traversable*, *rigid* and use no extension.

Code	Dungeon ¹⁷	Palace	Notes
ARCH_TOP_MID			This should be placed above an ARCH_BOTTOM and at left of an ARCH_TOP_LEFT, at right of an ARCH_TOP_RIGHT, or next to an ARCH_TOP_SMALL.
ARCH_TOP_SMALL			This should be placed in between two ARCH_TOP_MID.
ARCH_TOP_LEFT			This should be placed at left of an ARCH_TOP_RIGHT.

¹⁶ *Arches* are called *lattices* in the original game.

¹⁷ In the original game there is no dungeon version of arches.











ARCH_TOP_RIGHT



This should be placed at right of an ARCH_TOP_LEFT.

1.3 Backgrounds

Any combination of construction background and foreground is allowed, but be aware that some may simply not look right, or even not make a difference at all (like a torch behind a wall).

Code	Dungeon	Palace	Notes
NO_BRICKS			For the dungeon and palace this results in no graphics at all.
NO_BG			For the dungeon this results in no graphics. For the palace this results in the same graphics as BRICKS_02.
BRICKS_00			For the palace this results in no graphics.
BRICKS_01			For the palace this results in the same graphics as BRICKS_03
BRICKS_02			For the palace this results in the same graphics as NO_BG.
BRICKS_03			For the palace this results in the same graphics as BRICKS_01.
TORCH ¹⁸			






¹⁸ The fire sprites are mirrored horizontally depending on the level coordinates the torch has. This is not so in the original game.

WINDOW

BALCONY¹⁹

1.4 Items

The item extension is used only by FLOOR. Items are placed over the floor construction they belong to and animated. They can be caught by the kid by pressing the shift key.

Code	Graphics	Notes
NO_ITEM		This results in no graphics at all.
EMPTY_POTION		This potion has no bubbles and no effect.
SMALL_LIFE_POTION		This potion has red bubbles and recovers one life point.
BIG_LIFE_POTION		This potion has red bubbles, adds one life point container and recover all life points.
SMALL_POISON_POTION		This potion has blue bubbles and takes one life point. It has the same graphics as ACTIVATION_POTION.
BIG_POISON_POTION ²⁰		This potion has blue bubbles and takes all life point, effectively killing the kid.

¹⁹ The background stars are randomly positioned and animated. In the original game, all balconies have non-animated white stars in fixed positions.

²⁰ The big poison potion is not available in the original game.

FLOAT_POTION



This potion has green bubbles and allows the kid to levitate, effectively making long falls harmless.

FLIP_POTION



This potion has yellow bubbles and flips the screen horizontally, vertically or both, depending on its level coordinates.²¹ Drinking another one undo this effect.

ACTIVATION_POTION



This potion has the same graphics as `SMALL_POISON_POTION` and is used to activate a construction.²²

SWORD²³

This allows the kid to enter fight mode. The sword blinks from time to time.

²¹ In the original game, flip potions only invert the screen vertically and are drawn as `FLOAT_POTION`, that is, with green bubbles.

²² In the original game the *activation potion* is designed to be used in the potion level, a copy protection mechanism employed to stop people from sharing. **MININIM is free software**, and we repudiate this kind of measure. Thus here *activation potions* may be used to handicap the kid, but certainly won't be used to do so to you nor to your friends.

²³ The sword graphics is mirrored horizontally depending on its level coordinates. This is not so in the original game.

2 Movements

In his journey to save the princess, the kid has to endure several levels full of challenging traps and skilled opponents. Run, walk, turn, crouch, jump, hang, climb, take the sword, attack and defend are some of the kid's movements at your disposal to make the way to the princess tower. Although, this is a relatively large set of movements, basically all these are accomplished by some combination of the directional and shift keys, often conditioned to the current kid's state. This chapter explains in detail all possible movements of the kid.

There are six modes of movement: *normal*, *crouch*, *run*, *hang*, *fall* and *fight*. The mode the kid is in is the foremost element determining how the input is translated into kid's movements.

Notice that (whenever it makes sense) for any move, holding its triggering keys continues or repeats the movement.¹ Often, when transitioning from a movement to another that share some keys, you can keep the common keys pressed, as well as any other keys ignored by an intermediate movement, and the kid will just do the right thing. It's also worth nothing that the kid responds to the input in real time, that is, in the next animation cycle.²

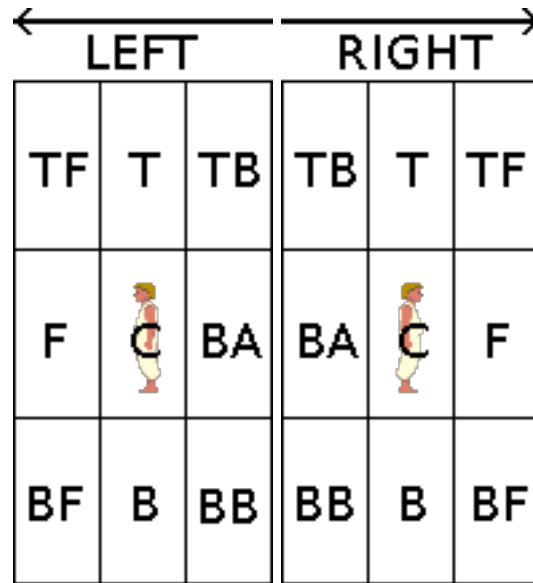
FRONT and *BACK* are meta-keys used to refer to *LEFT* and *RIGHT* conditionally subject to kid's facing direction. The *UP* and *DOWN* directional keys are simply referred as such.

¹ That's not so for the original game, in which for many situations you have to release and press the keys again to have the movement repeated.

² That's not so for the original game, in which the kid delays one cycle to respond.

2.1 Directed relative positioning

The concept of *directed relative positioning*, necessary to easily describe the relations between constructions and the kid is analogous to the already defined concept of *non-directed relative positioning*, used to describe relations between constructions. Consider the kid's place C , which has the coordinates (r, f, p) , its 8 adjacent places and the direction the kid is facing, as shown in the following two figures.



TF (top front) **T** (top) **TB** (top back) **F** (front) **C** (center)
BA (back) **BF** (bottom front), **B** (bottom) **BB** (bottom back)

Then we define the following mapping between names and coordinates:

Left	Right	Coordinates
TF	TB	$(r, f - 1, p - 1)$
T	T	$(r, f - 1, p)$
TB	TF	$(r, f - 1, p + 1)$
F	BA	$(r, f, p - 1)$
C	C	(r, f, p)
BA	F	$(r, f, p + 1)$
BF	BB	$(r, f + 1, p - 1)$
B	B	$(r, f + 1, p)$
BB	BF	$(r, f + 1, p + 1)$

2.2 Hangable place

Quite a few movements might culminate in the kid hanging on a construction. In order to describe the conditions in which that may happen, it's necessary to define the concept of *hangable place*.

Hangable place: we say the kid's place is *hangable*, if (and only if) its top construction is *strictly traversable* and its top front one is *hangable* in the opposite direction.

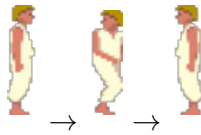
2.3 Normal

The movement mode at the beginning of each level is *normal*. While in it the kid is standing still.



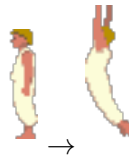
2.3.1 Turn

If you press *BACK*, the kid turns and then faces the opposite direction likewise.



2.3.2 Vertical jump

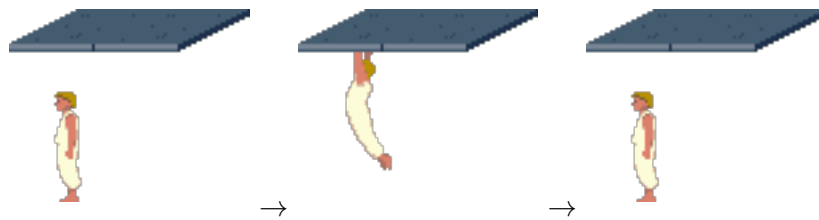
Pressing *UP*, makes the kid perform a vertical jump.



In case his *top construction* is *strictly traversable* but he isn't in a *hangable place*, he performs a long vertical jump and comes back to the **exact** same position he was before the jump.¹

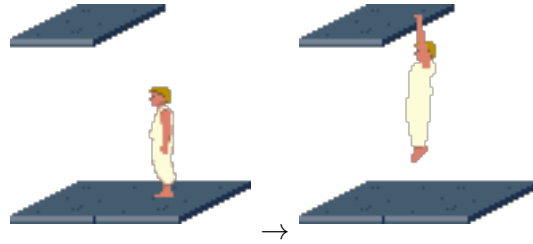


In case his *top construction* is not *strictly traversable* and his *back place* is not *hangable*, he performs a short vertical jump and then comes back to the exact same position he was before the jump.

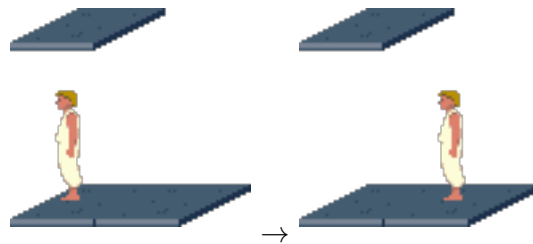


¹ This is not so in the original game, where he moves a little bit forward at each vertical jump.

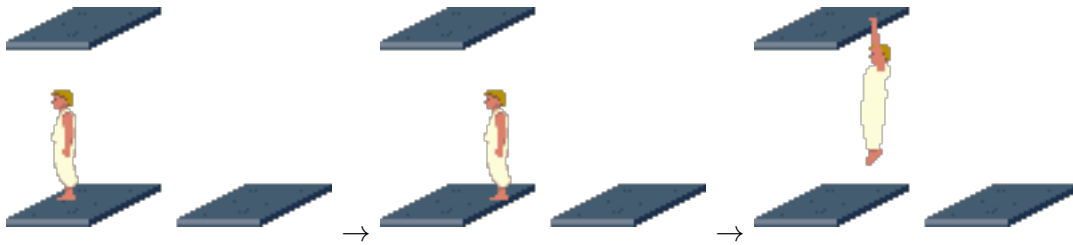
In case he's in a *hangable place*, he performs a long vertical jump (after coming close enough to the edge if he isn't already) and hangs for a brief period of time.



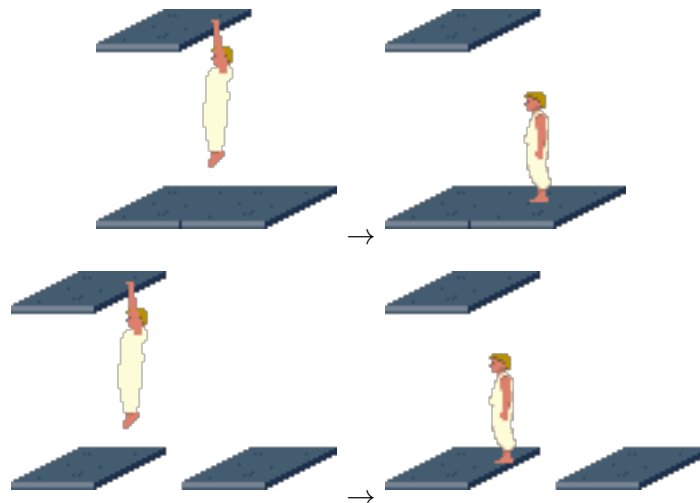
In case his *back place* is *hangable* and *non-strictly traversable*, he backs off so the situation fits the previous paragraph description.



In case his *back place* is *hangable* but *strictly traversable*, he backs off just enough, without entering the *strictly traversable* place, to perform a very vertical jump, and then hangs as described.

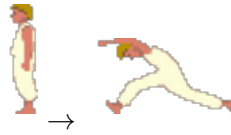


If holding *SHIFT*, the kid enters *hang mode*, otherwise he just drops back to the ground.

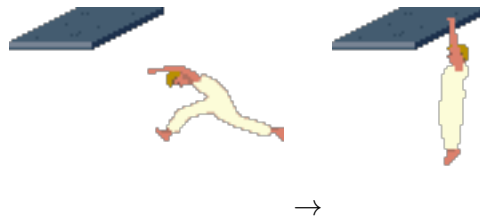


2.3.3 Normal jump

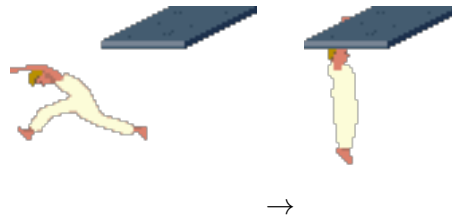
Pressing *UP* and *FRONT* at once makes the kid jump forward. He can jump over (up to) two *places* across without falling.



In midair, if he's in a *hangable place*, holding *SHIFT* and pressing *FRONT* makes him hang on the construction and enter *hang mode*.²

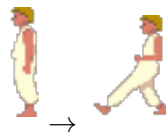


If he's in a *hangable place* for the opposite direction, holding *SHIFT* and pressing *BACK* makes him turn in midair, hang on the construction and enter *hang mode*.³

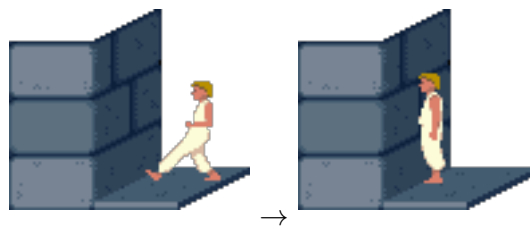


2.3.4 Walk

Pressing *FRONT* while holding *SHIFT*, makes the kid walk carefully.

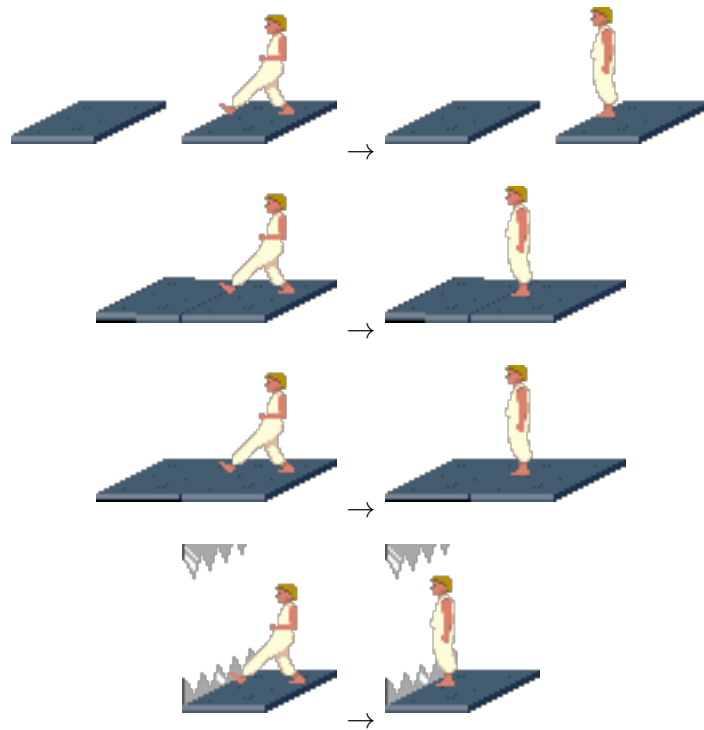


He always safely stops before a collision, fall, loose floor, closer floor and chopper blade.

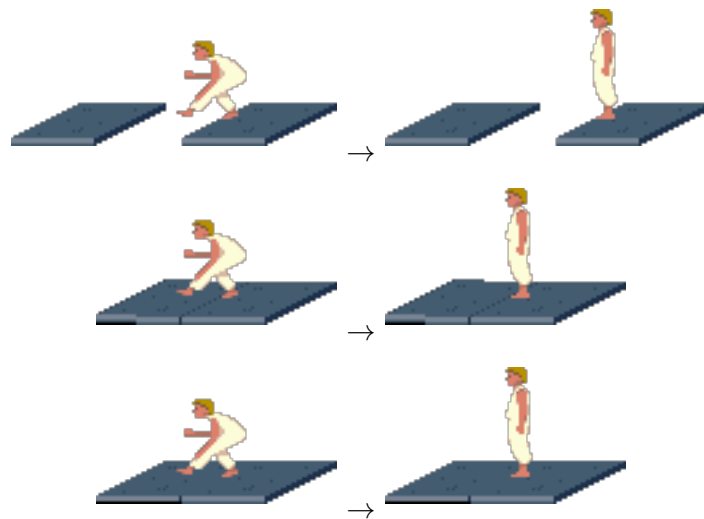


² The original game has no such movement.

³ The original game has no such movement.



If the kid is just before a *traversable* construction or a closer floor⁴, he misstep and gets back to his original position, as a warning of the eminent danger.

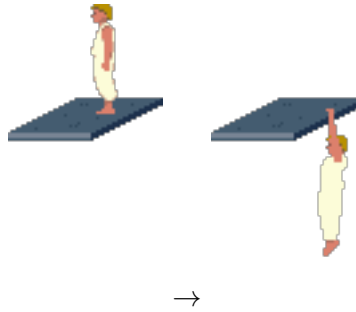


If you insist, he'll obey and suffer the consequences — so, be careful!

2.3.5 Descend

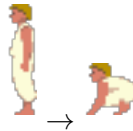
If you press *DOWN* and the kid's *bottom back place* is *hangable* and he's near enough to it, he descends, then hangs briefly — entering *hang mode* if *SHIFT* is kept pressed, or dropping off otherwise.

⁴ The kid doesn't misstep for a closer floor in the original game.

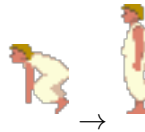


2.4 Crouch

If you press *DOWN* and the kid's *bottom back place* is *non-hangable*, or the kid is not near enough to it, he crouches.

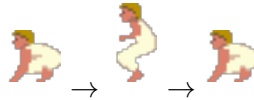


The kid stays crouched as long as you keep *DOWN* pressed. Releasing it makes him stand up again.



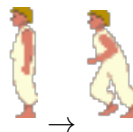
2.4.1 Crouching jump

If you press *FRONT* without releasing *DOWN*, the kid performs little crouching jumps forward.

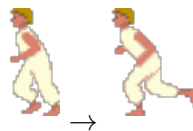


2.5 Run

Pressing *FRONT*, makes the kid start to run.⁵

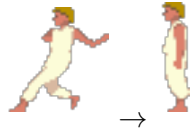


He runs as long as you keep *FRONT* pressed.



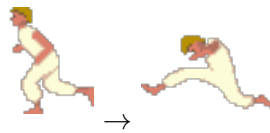
⁵ Unless he is facing a *collidable* construction with at most one place of distance, in which case he performs a *walk*. Notice that in the original game the threshold is lower.

Releasing it makes him stop, and go back to the normal stage.



2.5.1 Running jump

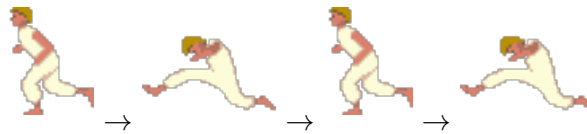
In case you press *UP* while running, the kid performs a running jump. The running jump has a reach greater than the *normal jump*. The kid can jump over (up to) three *places* across without falling.



When the kid lands on ground, he continues to run if you keep *FRONT* pressed.



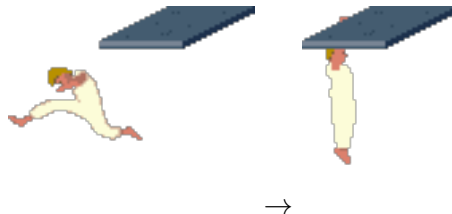
This implies that, if you keep both keys pressed the movement repeats indefinitely.



In midair, if he's in a *hangable place*, holding *SHIFT* and pressing *FRONT* makes him hang on the construction and enter *hang mode*.⁶



If he's in a *hangable place* for the opposite direction, holding *SHIFT* and pressing *BACK* makes him turn in midair, hang on the construction and enter *hang mode*.⁷

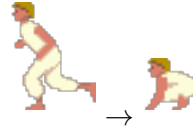


⁶ The original game has no such movement.

⁷ The original game has no such movement.

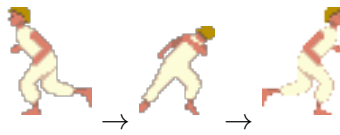
2.5.2 Running crouch

If you press *DOWN* while running, he crouches, slipping with some momentum, and then he enters *crouch mode*.



2.5.3 Running turn

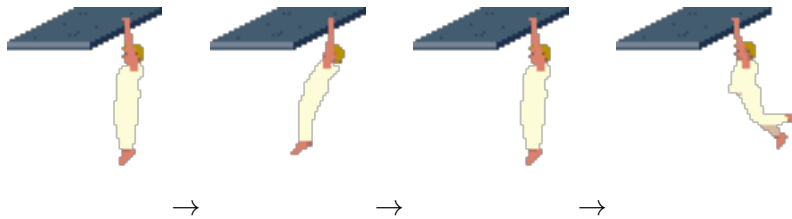
If you press *BACK* while running, the kid turns without stopping and continues to run in the opposite direction if you keep it pressed.



2.6 Hang

There are two modes of hanging: *free* and *non-free*. The kid *hangs free* when his *front construction* is not collidable in the direction of hanging. Otherwise he *hangs non-free*.

While *hanging free* the kid swings back and forth like a pendulum.



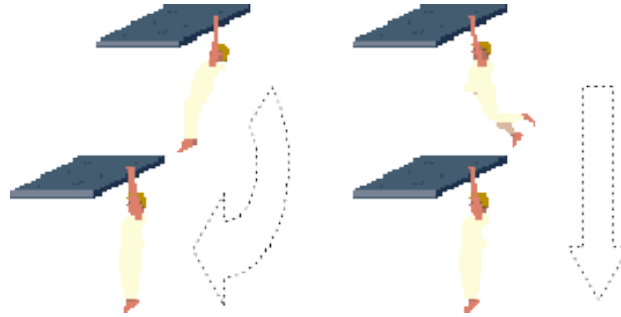
By contrast, when *hanging non-free* the kid keeps himself static.



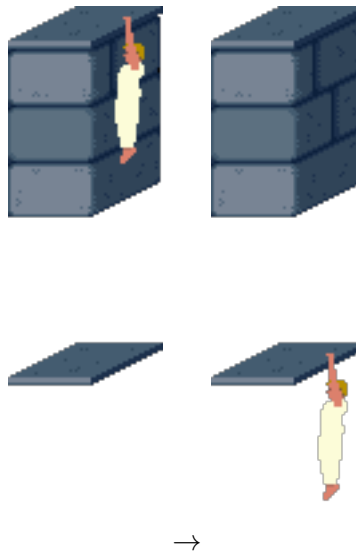
2.6.1 Hanging fall

When *hanging free*, there is a limit for how much time the kid can support his own weight, after which he drops from the construction. Releasing *SHIFT* in the meantime has the same effect. The kid will fall either aligned with the *hangable place's column* or with the *hangable place's front column*, depending upon which his feet are in when he leaves the hanged construction.⁸

⁸ It's not possible to perform such movements in the original game, because there the kid always falls over a *non-strictly traversable* construction, in case there is one.

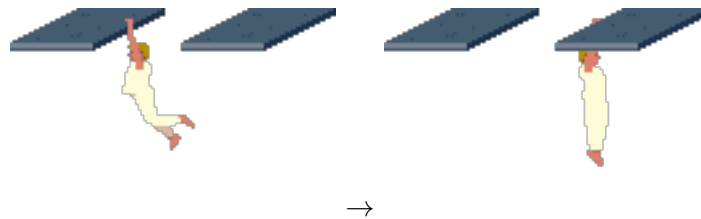


The kid can sustain himself indefinitely when *hanging non-free*, thus the only way to make him drop is by releasing *SHIFT*. As one would expect, the kid always falls aligned to the *hangable place's column*.



2.6.2 Hanging turn

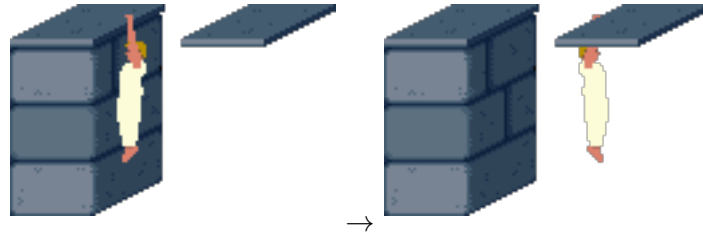
When the kid is *hanging free* in a place that is *hangable* in both directions and his feet are in the *hangable place*, pressing *BACK* while holding *SHIFT*, makes him turn in midair and hang on the con at his back.⁹



Naturally, if *hanging non-free* in a place that is *hangable* in both directions, there is no need to observe the kid's body inclination, just pressing *BACK*, will do the job (granted *SHIFT* is kept pressed).¹⁰

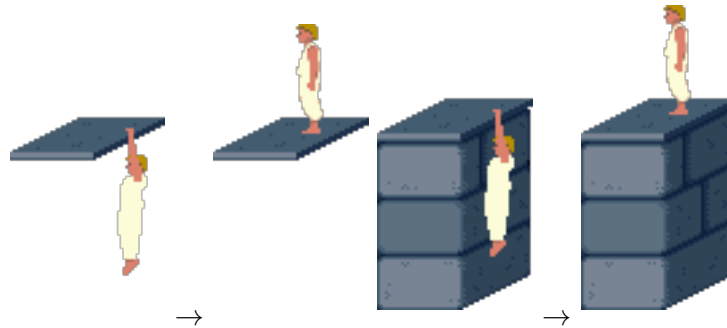
⁹ This movement is not present in the original game.

¹⁰ This movement is not present in the original game.



2.6.3 Climb

Pressing *UP* while hanging makes the kid climb the construction for both modes.



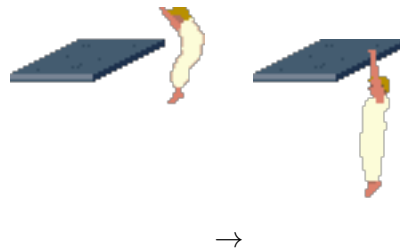
2.7 Fall

This mode is not really related to a key combination, but instead to a circumstance: the kid trying to support his own weight on an *strictly traversable* construction. When this happens he accelerates downwards (eventually reaching terminal velocity), until he encounters a *non-strictly traversable* construction or hangs at a *hangable place* which might appear in his way down.

If the kid hits a *non-traversable* construction with sufficient speed, he is harmed or can even die. Falling from a height of one floor inflicts no damage on him. Two floors takes one life point. Three or more kills him instantly.

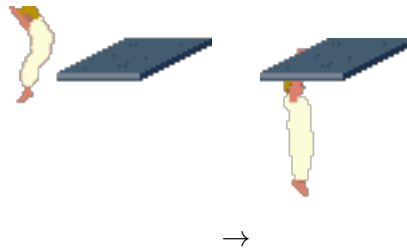
2.7.1 Falling hang

When falling if the kid is passing through a *hangable place* and his vertical velocity is low enough, and he's near enough to the *hangable construction*, he is able to hang there and enter *hang mode* if you hold *SHIFT* down.



2.7.2 Falling turn

If instead the place is *hangable* at the opposite direction, holding *SHIFT* and pressing *BACK* makes the kid turn in mid air, hang there and enter *hang mode*.¹¹



2.8 Fight

If the kid has got the sword, *RETURN* takes the sword and puts him in *fight mode*.¹²



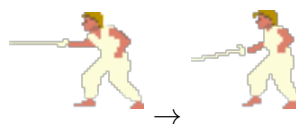
2.8.1 Fight walk forward

Pressing *FRONT*, makes the kid walk forward.



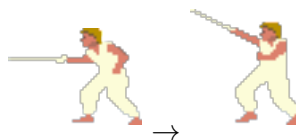
2.8.2 Fight walk backward

If you press *BACK*, the kid goes backward.



2.8.3 Defense

If you press *UP*, the kid makes the defense movement.

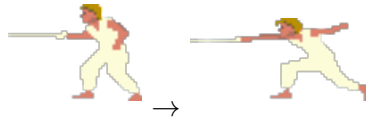


2.8.4 Attack

Pressing *SHIFT* makes the kid attack.

¹¹ This movement is not available in the original game.

¹² In the original game, the kid enters fight mode automatically in the eminence of an enemy threat.



2.8.5 Counter attack and counter defense

Holding *SHIFT* and *UP* puts the kid in counter attack and counter defense mode. At first, there is no immediate noticeable difference from standing still in fight mode, but as an enemy attacks¹³, the kid defends for certain. Then, depending upon probability whose likelihood is given by the KCA (Kid’s Counter Attack skill), he may counter attack.

If he does, depending on the enemy’s skills, they may counter defend and then counter attack. In that case, depending upon probability whose likelihood is given by the KCD (Kid’s Counter Defense skill), the kid may counter defend and the process restarts at the kid’s counter attack stage, making for a chain of counter attacks and counter defenses whose length is determined only by the fighters’ skills (and a little bit of luck). Every counter attack that is not responded with a counter defense by the opponent, makes him lose a life point.

When standing still in this mode, releasing *UP* initiates an attack (but at the same time makes the kid briefly vulnerable to the enemy’s attacks). In this case the same counter attack and counter defense chain logic described above is symmetrically applied.

Pressing *FRONT* and *BACK* (while holding *SHIFT* and *UP*) makes the kid walk forward and backward, respectively, without leaving this mode. However, although walking backward doesn’t make him lower his guard, walking forward makes him vulnerable until he puts himself on guard again.

The kid’s skills are built up by defeating enemies.¹⁴ The more advanced the defeated enemy’s skills, the more the kid’s skills improve. Both skills —KCA and KCD— are given by a number between 0 and 100, interpreted as percentages, indicating the likelihood of counter attacking and counter defending, respectively.

2.8.6 Keep sword

Finally you can make the kid keep his sword and return to *normal mode* by pressing *DOWN*.



¹³ In the original game the terminology for “attack” is “striking”, for “counter attack” is “re-striking after block”, for “defense” is “blocking strike” and for “counter defense” is “improper blocking”.

¹⁴ In the original game there is no such skills scheme. There, if you don’t get tired of pressing *SHIFT* and *UP*, the kid’s counter attack and counter defense are infallible.

3 Key bindings

The engine has several key bindings that control an wide range of settings and perform useful functions. Many key bindings work anytime, in the title screen, cut scenes and levels, while others work only in the latter. The key bindings are designed to mimic those of the original game when started with the option `megahit` (improved in latter revisions), however they can implement a super-set of the functions available there and MININIM implements quite a few more. Also, many key bindings have a configuration option counterpart. [Chapter 4 \[Configuration\]](#), page 30.

3.1 Game flow

- SHIFT+L* Warp to next level. Related to option `--start-level`.
- CTRL+A* Restart level.
- CTRL+R* Restart the game.
- CTRL+Q* Quit the game.

3.2 Graphical rendering

These key bindings modify how the engine graphically renders the game world objects.

- F10* Select guard mode. This circulates around the modes: “ORIGINAL”, “GUARD”, “FAT GUARD”, “VIZIER”, “SKELETON” and “SHADOW”. The “ORIGINAL” mode gives level modules autonomy in this choice for each particular guard. This is the default. Related to option `--guard-mode`.
- F11* Select environment mode. This circulates around the modes: “ORIGINAL”, “DUNGEON” and “PALACE”. The “ORIGINAL” value gives level modules autonomy in this choice for each particular level. This is the default. Related to option `--environment-mode`.
- F12* Select video mode. This circulates around the modes: “VGA”, “EGA”, “CGA” and “HGC”. The default is VGA. Related to option `--video-mode`.
- SHIFT+B* Toggle blind mode. In blind mode background and non-animated sprites are not drawn. The default is to draw everything. Related to option `--blind-mode`.

3.3 Life and death

- K* Kill enemy.¹
- R* Resurrect kid.
- I* Toggle immortal mode. In immortal mode the kid can't be harmed. The default mode is mortal. Related to option `--immortal-mode`.
- SHIFT+S* Increment kid's current life points.
- SHIFT+T* Increment kid's total life points. Related to option `--total-lives`.

¹ This only works for the enemy currently associated with the kid. In the original game this works for the enemy in the room at view, as it can only handle one guard per room. There is no such limit in MININIM.

3.4 Kid special

These key bindings give or trigger special kid's abilities.

- A* Alternate between kid and its shadows. This is intended for level modules that make use of such feature. That's not the legacy module case.
- SHIFT+W* Float kid. This is equivalent to drinking a float potion.

3.5 Time

These key bindings allow you to control the remaining time available to end the game. They are related to options `--time-limit` and `--start-time`.

- SPACE* Display remaining time. Notice that the remaining time is displayed automatically when it reaches multiples of 5 minutes, and it's constantly shown when remains less than a minute to its end.
- +* Increment and display remaining time. If the remaining time is greater than 1 minute, increment remaining time by 1 minute, otherwise by 1 second.
- Decrement and display remaining time. If the remaining time is greater than 1 minute, decrement remaining time by 1 minute, otherwise by 1 second.

3.6 Skills

These key bindings allow you to set the kid skills. They are related to options `--kca` and `--kcd`.

- TAB* Display skills. If the kid is associated with an enemy, it shows both fighters' skills, otherwise it only shows the kid's.
- CTRL+=* Increment KCA skill.
- CTRL+-* Decrement KCA skill.
- ALT+=* Increment KCD skill.
- ALT+-* Decrement KCD skill.

3.7 Level navigation

These key bindings allow you to move the camera around the level, making it possible to view any room linked to the current one without having to move the kid there. Notice that if the kid enters the room the camera is on, it starts following him again. There is also a couple of key bindings intended to show room coordinates, which is useful when building levels.

- H* Move the camera to the room at left (right if the screen is flipped horizontally).
- J* Move the camera to the room at right (left if the screen is flipped horizontally).
- U* Move the camera to the room above (below if the screen is flipped vertically).
- N* Move the camera to the room below (above if the screen is flipped vertically).
- HOME* Move the camera back to the kid's room.
- C* Show direct coordinates.
- SHIFT+C* Show indirect coordinates.

3.8 Files

These key bindings are used to save and load configuration files. They open a platform-dependent file choosing dialog. Notice that save files are a particular case of configuration files and should be loaded the same way. When loaded in-game, configuration files take effect immediately, however some effects only become visible under certain conditions. For instance save files show their effect only on game (re)start.

CTRL+L Load configuration. Related to option `--load-config`.

CTRL+G Save game.

3.9 Peripherals

These key bindings allow you to tune hardware peripherals that work with the game.

F Toggle fullscreen mode. In fullscreen mode the window spans the entire screen. The default is to have a window of 640x400 resolution. Related to option `--fullscreen`.

SHIFT+I Select display flip mode. This circulates around the modes: “NONE”, “VERTICAL”, “HORIZONTAL” and “VERTICAL + HORIZONTAL”. The default is “NONE”. Related to option `--display-flip-mode`.

SHIFT+K Select keyboard flip mode. This circulates around the modes: “NONE”, “VERTICAL”, “HORIZONTAL” and “VERTICAL + HORIZONTAL”. The default is “NONE”. Related to option `--keyboard-flip-mode`.

CTRL+S Toggle sound. The default is to have sound enabled. Related to option `--sound`.

3.10 Miscellaneous

These are key bindings that don't fit in any other category.

CTRL+V Show engine name and version. Related to option `--version`.

4 Configuration

MININIM can be easily setup to match user's preferences regarding the way the game looks and behaves. It features an integrated configuration system which offers three distinct interfaces: *command line options*, *environment variables* and *configuration files*.

The configuration system uses the command line options parser as the central interpreter for the configuration process. Providing a single options parsing routine allows for easy maintenance, equivalence between methods and a canonical interface for integrating future methods.

For that end, each configuration option backend implements a translator between its specifics of option representation and access, to a normal command line array. Then the resulting options are roughly processed as if they were given at the engine invocation command line.

As the options are the same for all three current methods of configuration, and all of them are in essence processed as command line options, it's unnecessary to describe all options for each separately. Therefore, we provide systematic rules for converting between method option names and their command line equivalents, and then describe only the latter.

The rule for option naming conversion is very simple: for every command line option of the form `x-y` there is an equivalent environment variable option `MININIM_X_Y` and an equivalent configuration file option `'x y'`.

The multiple methods of configuration are cumulative but command line options override any other, while environment variables override the main configuration file. In any method applicable later options override earlier ones. The option `--print-paths` shows, among other things, the expected file name of the main configuration file. Configuration files have INI format without any sections, like this:

```
option 0 = value 0
...
option n = value n
```

Notice that many options can be changed on the fly by key binding counterparts. See [Chapter 3 \[Key bindings\], page 27](#).

The following remarks apply to the options list in this chapter:

- Long option names are case sensitive.
- Option values are case insensitive.
- Both can be partially specified as long as they are kept unambiguous.
- *boolean* is an integer equating to '0', or any sub-string (including the null string) of 'FALSE', 'OFF' or 'NO' to disable the respective feature, and any other value (even no string at all) to enable it.
- For any non-specified option the documented default applies.
- Integers can be specified in any of the formats defined by the C language.
- Key bindings references are based on the default mapping.

4.1 Configuration options

`--ignore-environment`

Ignore environment variables. The default is to parse them after the main configuration file.

`--ignore-main-config`

Ignore main configuration file. The default is to parse it at the very beginning of each run.

`--load-config=FILE`

Load configuration file *file*. The options set in *file* have the same precedence as the equivalent command line options given at its place of occurrence. This can be done in-game by the `CTRL+L` key binding.

4.2 Level options

`--level-module=LEVEL-MODULE`

Select level module. A level module determines a way to generate consecutive levels for use by the engine. Valid values for *level-module* are: ‘LEGACY’ and ‘CONSISTENCY’. ‘LEGACY’ is the module designed to read the original unarchived PoP (Prince of Persia) 1 DOS level files. ‘CONSISTENCY’ is the module designed to generate random-corrected levels for accessing the engine robustness. The default is ‘LEGACY’.

`--start-level=N`

Make the kid start at level *n*. The default is ‘1’. Valid integers range from ‘1’ to `INT_MAX`. This can be changed in-game by the `SHIFT+L` key binding.

4.3 Time options

`--start-time=N`

Set the play time counter to *n* seconds. The default is ‘0’. Valid integers range from ‘0’ to `INT_MAX`.

`--time-limit=N`

Set the time limit to complete the game to *n* seconds. The default is ‘3600’ (1 hour). Valid integers range from ‘1’ to `INT_MAX`. This can be changed in-game by the `+` and `-` key bindings.

4.4 Skills options

`--immortal-mode[=BOOLEAN]`

Enable/disable immortal mode. In immortal mode the kid can’t be harmed. The default is ‘FALSE’. This can be changed in-game by the `I` key binding.

`--kca=N` Set kid’s counter attack skill to *n*. The default is ‘0’. Valid integers range from ‘0’ to ‘100’. This can be changed in-game by the `CTRL+=` and `CTRL+-` key bindings.

- `--kcd=N` Set kid's counter defense skill to *n*. The default is '0'. Valid integers range from '0' to '100'. This can be changed in-game by the *ALT+=* and *ALT+-* key bindings.
- `--total-lives=N` Make the kid start with *n* total lives. The default is '3'. Valid integers range from '1' to '10'. This can be changed in-game by the *SHIFT+T* key binding.

4.5 Rendering options

- `--blind-mode[=BOOLEAN]`
Enable/disable blind mode. In blind mode background and non-animated sprites are not drawn. The default is 'FALSE'. This can be changed in-game by the *SHIFT+B* key binding.
- `--display-flip-mode=DISPLAY-FLIP-MODE`
Select display flip mode. Valid values for *display-flip-mode* are: 'NONE', 'VERTICAL', 'HORIZONTAL' and 'VERTICAL-HORIZONTAL'. The default is 'NONE'. This can be changed in-game by the *SHIFT+I* key binding.
- `--environment-mode=ENVIRONMENT-MODE`
Select environment mode. Valid values for *environment-mode* are: 'ORIGINAL', 'DUNGEON' and 'PALACE'. The 'ORIGINAL' value gives level modules autonomy in this choice for each particular level. This is the default. This can be changed in-game by the *F11* key binding.
- `--guard-mode=GUARD-MODE`
Select guard mode. Valid values for *guard-mode* are: 'ORIGINAL', 'GUARD', 'FAT-GUARD', 'VIZIER', 'SKELETON' and 'SHADOW'. The 'ORIGINAL' value gives level modules autonomy in this choice for each particular guard. This is the default. This can be changed in-game by the *F10* key binding.
- `--mirror-mode[=BOOLEAN]`
Enable/disable mirror mode. In mirror mode the screen and the keyboard are flipped horizontally. This is equivalent of specifying both the options `--display-flip-mode=HORIZONTAL` and `--keyboard-flip-mode=HORIZONTAL`. The default is 'FALSE'. This can be changed in-game by the *SHIFT+I* and *SHIFT+K* key bindings for the display and keyboard, respectively.
- `--video-mode=VIDEO-MODE`
Select video mode. Valid values for *video-mode* are: 'VGA', 'EGA', 'CGA' and 'HGC'. The default is 'VGA'. This can be changed in-game by the *F12* key binding.

4.6 Window options

- `--fullscreen[=BOOLEAN]`
Enable/disable fullscreen mode. In fullscreen mode the window spans the entire screen. The default is 'FALSE'. This can be changed in-game by the *F* key binding.

- `--window-dimensions=WxH`
Set window width and height to *w* and *h*, respectively. The default is '640x400'. The values *w* and *h* are strictly positive integers and must be separated by an 'x'.
- `--window-position=X,Y`
Place the window at screen coordinates *x,y*. The default is to let this choice to the window manager. The values *x* and *y* are integers and must be separated by a comma.

4.7 Paths options

- `--data-path=PATH`
Set data path to *path*. Normally, the data files are looked for in the current working directory, then in the user data directory, then in the resources directory, and finally in the system data directory. If this option is given, before looking there the data files are looked for in *path*.
- `--print-paths`
Print paths and exit.

4.8 Other options

- `--inhibit-screensaver[=BOOLEAN]`
Prevent the system screensaver from starting up. The default is 'FALSE'.
- `--keyboard-flip-mode=KEYBOARD-FLIP-MODE`
Select keyboard flip mode. Valid values for *keyboard-flip-mode* are: 'NONE', 'VERTICAL', 'HORIZONTAL' and 'VERTICAL-HORIZONTAL'. The default is 'NONE'. This can be changed in-game by the *SHIFT+K* key binding.
- `--skip-title[=BOOLEAN]`
Skip title screen. The default is 'FALSE'.
- `--sound[=BOOLEAN]`
Enable/disable sound. The default is 'TRUE'. This can be changed in-game by the *CTRL+S* key binding.

4.9 Help options

- `-?`
- `--help` Print options list and exit
- `--usage` Print a short usage message and exit
- `-V`
- `--version` Print program name and version and exit

5 Software package

This chapter briefly discuss matters relevant to MININIM while a software package: build, installation, distribution and contribution are some of the covered topics. Bear in mind that instructions in this chapter are package-specific; for general and in-depth configuration, build and installation instructions refer to the `INSTALL` file present in the top-level directory of the source distribution. If you have checked out the source tree from the VCS repository see [Section 5.8 \[Hacking\], page 35](#).

For more information about this program you can visit its home page at <http://oitofelix.github.io/mininim/>. If you want to receive notifications about new releases of this program subscribe to the author's atom feed at <http://oitofelix.github.io/feed.xml>.

5.1 Description

MININIM is the Advanced Prince of Persia Engine — a childhood dream, the free software engine for Jordan Mechner's masterpiece, developed from scratch by Bruno Félix Rezende Ribeiro ([oitofelix](#)).

MININIM aims to have detailed documentation on all matters concerning the original 1989 Prince of Persia world, ranging from its design concepts to the engine's practical use and development.

5.2 Distribution

This program is *free software*; this means that everyone is free to use it and free to redistribute it under certain conditions. This program is not in the public domain; it is copyrighted and there are restrictions on its distribution, but these restrictions are designed to permit everything that a good cooperating citizen would want to do. What is not allowed is to try to prevent others from further sharing any version of this program that they might get from you. The precise conditions are found in the GNU [GPL](#) (General Public Licence). The program manual is covered by the GNU Free Documentation License ([Appendix A \[GNU FDL\], page 40](#)). This license is similar in spirit to the GNU General Public License, but is more suitable for documentation.

5.3 Getting a copy

One way to get a copy of this program is from someone else who has it. You need not ask for our permission to do so, or tell any one else; just copy it. If you have access to the Internet, you can get the latest distribution version of this program at <http://oitofelix.github.io/mininim/>.

A VCS repository, where the development takes place, is also available. It maintains the full history of modifications for every single source file. You can checkout any given revision of any file or get an snapshot of the entire source tree in a particular desired state. Special build tools, as described in [Section 5.8 \[Hacking\], page 35](#), are required to build from those checkouts. To stay up to date with the latest developments in the source tree, you can anonymously checkout the repository with the following command:

```
git clone https://github.com/oitofelix/mininim.git
```

5.4 Contact

You can get in touch with other users of this program by posting to the forum at <http://forum.princed.org/index.php>. You can contact the author at oitofelix@gnu.org. You may also join us on our friendly IRC channel at <irc://irc.freenode.net/mininim>.

5.5 Bug reporting

You can report bugs at <https://github.com/oitofelix/mininim/issues>. Please, when reporting a bug include enough information for the author to reproduce the problem. Generally speaking, that means:

- The contents of any input files necessary to reproduce the bug and command line invocations of the program(s) involved (crucial!).
- A description of the problem and any samples of the erroneous output.
- The version number of the program(s) involved (use `--version`).
- Hardware, operating system, and compiler versions (`uname -a`).
- Unusual options you gave to configure, if any (see `config.status`).
- Anything else that you think would be helpful.

5.6 Contributing

This program is a collaborative effort and we encourage contributions from anyone and everyone — your help is very much appreciated. You can help in many ways:

- Donate to the author in order to support his work. See [Section 5.7 \[Donating\]](#), page 35.
- Help users in the forum and IRC channel. See [Section 5.4 \[Contact\]](#), page 35.
- Find and report bugs. See [Section 5.5 \[Bug reporting\]](#), page 35.
- Suggest new features.
- Translate the program strings to other languages.

5.7 Donating

MININIM development is voluntary. If you like this game and want to thank the author for the work he has done so far, or want to enable him to spend more time improving it, please make a donation! For values equal or greater than 50 US\$ or 0.12 BTC your name will be acknowledged in this manual ([Section 6.5 \[Donors\]](#), page 39) and in the donors page at the author's website (<http://oitofelix.github.io/funding.html#donors>). If you decide to make recurrent donations, the total amount of your donations will be updated every month for all records. Please, donate at <http://oitofelix.github.io/funding.html>.

Thank you for your support!

5.8 Hacking

The development sources are available through VCS:

<https://github.com/oitofelix/mininim>

If you are getting the sources from the VCS (or change `configure.ac`), you'll need to have Automake, Autoconf and Gettext installed to (re)build. You'll also need `help2man`.

After getting the VCS sources, and installing the tools above, you can run `./bootstrap && ./configure && make` to do a fresh build. After that first time, running `make` should suffice to rebuild the program with your changes. See file `INSTALL`.

When modifying the sources, or making a distribution, more is needed, as follows:

- This distribution also uses `Gnlib` to share common files, stored as a submodule in git.
- When updating gettext, besides the normal installation on the system, it is necessary to run `gettextize -f` in this hierarchy to update the `po/` infrastructure. After doing so, rerun `gnlib-tool --import` since otherwise older files will have been imported. See *Gnlib Manual*, for more information.

When committing changes to the repository always create an entry in the `doc/release/latest-news.texi` file for any user-visible changes or additions made. This file is intended to provide the latest release news for the `NEWS.texi` and `ANNOUNCEMENT.texi` files to avoid duplication of information and syncing work. After a release is made the news items should be moved to the `NEWS.texi` file and another news list should be built from scratch in the `ANNOUNCEMENT.texi` file.

5.9 MinGW

These are the steps to build using MinGW, in a machine running Microsoft Windows.

- Download and install git: <https://git-scm.com/download/win>.
- Clone the gnlib repository: `git clone git://git.savannah.gnu.org/gnlib.git` and add the gnlib root to your `PATH`.
- Download MinGW: <http://downloads.sourceforge.net/project/mingw/Installer/mingw-get-setup.exe>
- Install the packages: `'mingw32-base', 'mingw32-gcc-g++', 'msys-base', 'mingw32-autoconf', 'mingw32-automake'`.
- Rename the file `C:/MinGW/msys/1.0/etc/fstab.sample` to `fstab`.
- Download Allegro: <http://cdn.allegro.cc/file/library/allegro/5.0.10/allegro-5.0.10-mingw-4.7.0.zip> and extract its root folder's contents to `C:\MinGW`.
- Run `msys`: `C:\MinGW\msys\1.0\msys.bat`
- Clone the repository: `git clone https://github.com/oitofelix/minim.git`
- Build: `cd minim && ./bootstrap && ./configure && make`
- Distribute the resulting binary `minim.exe` along with the following dlls: `C:\MinGW\bin\libstdc++-6.dll` and `C:\MinGW\bin\libgcc_s_dw2-1.dll`.

6 Release

In this chapter one can find information that are specific to the current release or subject to change between releases. It presents the features this package acquired over time and what are our plans for it. Last but not least, people whose contribution for this package are noteworthy are acknowledged.

6.1 Announcement

MININIM is the Advanced Prince of Persia Engine — a childhood dream, the free software engine for Jordan Mechner’s masterpiece, developed from scratch by Bruno Félix Rezende Ribeiro ([oitofelix](https://github.com/oitofelix)).

MININIM aims to have detailed documentation on all matters concerning the original 1989 Prince of Persia world, ranging from its design concepts to the engine’s practical use and development.

News

- First release

Download

Here are the compressed sources and a GPG detached signature:

```
http://oitofelix.github.io/mininim/mininim-0.9.2.tar.gz
http://oitofelix.github.io/mininim/mininim-0.9.2.tar.gz.sig
```

Use the `.sig` file to verify that the corresponding file (without the `.sig` suffix) is intact. First, be sure to download both the `.sig` file and the corresponding tarball. Then, run this command:

```
gpg --verify mininim-0.9.2.tar.gz.sig
```

If that command fails because you don’t have the required public key, then run this command to import it:

```
gpg --recv-keys 0x28D618AF --keyserver hkp://keys.gnupg.net
```

and rerun the `gpg --verify` command.

This release is signed by *Bruno Félix Rezende Ribeiro*. His key fingerprint is 7CB1 208C 7336 56B7 5962 2500 27B9 C6FD 28D6 18AF.

This release was bootstrapped with the following tools:

- GNU Autoconf 2.69
- GNU Automake 1.14.1
- GNU Texinfo 5.2

Links

Homepage <http://oitofelix.github.io/mininim/>

Atom feed <http://oitofelix.github.io/feed.xml>

Forum <http://forum.princed.org/index.php>

IRC channel

`irc://irc.freenode.net/mininim`

Bug reporting

`https://github.com/oitofelix/mininim/issues`

VCS

`https://github.com/oitofelix/mininim`

Please, support MININIM development!

MININIM development is voluntary. If you like this game and want to thank the author for the work he has done so far, or want to enable him to spend more time improving it, please make a donation! For values equal or greater than 50 US\$ or 0.12 BTC your name will be acknowledged in this manual (Section 6.5 [Donors], page 39) and in the donors page at the author's website (<http://oitofelix.github.io/funding.html#donors>). If you decide to make recurrent donations, the total amount of your donations will be updated every month for all records. Please, donate at <http://oitofelix.github.io/funding.html>.

Thank you for your support!

6.2 News

This document contains a list of user-visible changes worth mentioning. The changes are split and ordered by version in reverse chronological order.

0.9.2

- First release

6.3 To do

This section contains a list of ideas and features the author plans to implement. If you would like to see any of them implemented, **please donate!** See Section 5.7 [Donating], page 35.

- Joystick support
- Native level format
- Level editor
- Level generator module
- Guile scripting
- Network play

6.4 Thanks

This section contains a list of people that submitted actual code. The names are listed in chronological order of contribution. If you would appreciate your own name listed here, **please contribute!** See Section 5.6 [Contributing], page 35.

—

6.5 Donors

This section contains a list of people that supported the development financially. This program would not be what it is today without the invaluable help of these people, to whom we would like to say:

THANK YOU VERY MUCH!

If you would appreciate your own name listed here, **please donate!** See [Section 5.7 \[Donating\]](#), page 35.

Name

Email

\$

—

—

—

Appendix A GNU FDL

Version 1.3, 3 November 2008

FSF (Free Software Foundation) (<http://fsf.org/>)
51 Franklin St., Floor 5
Boston, MA 02110-1335
USA

This document is part of GNU philosophy, the GNU Project's exhaustive collection of articles and essays about free software and related matters.

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

Verbatim copying and distribution of this entire appendix are permitted worldwide, without royalty, in any medium, provided this notice is preserved.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of

mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other

implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your

option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.3 or  
any later version published by the Free Software Foundation; with  
no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled  
‘‘GNU Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with the  
Front-Cover Texts being list, and with the Back-Cover Texts  
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix B Linux and the GNU system

The original version of this essay was published as the file `etc/LINUX-GNU` in the GNU Emacs distribution.

This document is part of GNU philosophy, the GNU Project's exhaustive collection of articles and essays about free software and related matters.

Copyright © 1996, 2002 Richard Stallman

Verbatim copying and distribution of this entire appendix are permitted worldwide, without royalty, in any medium, provided this notice is preserved.

The GNU project started in 1984 with the goal of developing a complete free Unix-like operating system: GNU. “Free” refers to freedom, not price; it means you are free to run, copy, distribute, study, change, and improve the software.

A Unix-like system consists of many different programs. We found some components already available as free software—for example, X Windows and \TeX . We obtained other components by helping to convince their developers to make them free—for example, the Berkeley network utilities. This left many missing components that we had to write in order to produce GNU—for example, GNU Emacs, the GNU C compiler, the GNU C library, Bash, and Ghostscript. The GNU system consists of all these components together.

The GNU project is not just about developing and distributing some useful free software. The heart of the GNU project is an idea: that software should be free, that software users should have freedom to participate in a community. To run your computer, you need an operating system; if it is not free, your freedom has been denied. To have freedom, you need a free operating system. We therefore set out to write one.

In the long run, though, we cannot expect to keep the free operating system free unless the users are aware of the freedom it gives them, and value that freedom. People who do not appreciate their freedom will not keep it long. If we want to make freedom last, we need to spread awareness of the freedoms they have in free software.

The GNU project's method is that free software and the idea of users' freedom support each other. We develop GNU software, and as people encounter GNU programs or the GNU system and start to use them, they also think about the GNU idea. The software shows that the idea can work in practice. Some of these people come to agree with the idea, and then they are more likely to write additional free software. Thus, the software embodies the idea, spreads the idea, and grows from the idea.

Early on in the development of GNU, various parts of it became popular even though users needed proprietary systems to run them on. Porting the system to many systems and maintaining them required a lot of work. After that work, most GNU software is easily configured for a variety of different platforms.

By 1991, we had found or written all of the essential major components of the system except the kernel, which we were writing.¹

That was the situation when Linux came into being. Linux is a kernel, like the kernel of Unix; it was written by Linus Torvalds, who released it under the GNU General Public

¹ This kernel consists of the Mach microkernel plus the GNU HURD. The first test release was made in 1996. Now, in 2002, it is running well, and Hurd-based GNU systems are starting to be used.

License. He did not write this kernel for GNU, but it fit into the gap in GNU. The combination of GNU and Linux included all the major essential components of a Unix-compatible operating system. Other people, with some work made the combination into a usable system. The principal use of Linux, the kernel, is as part of this combination.

The popularity of the GNU/Linux combination is success, in the sense of popularity, for GNU. Ironically, the popularity of GNU/Linux undermines our method of communicating the ideas of GNU to people who use GNU.

When GNU programs were only usable individually on top of another operating system, installing and using them meant knowing and appreciating these programs, and thus being aware of GNU, which led people to think about the philosophical base of GNU. Now users can install a unified operating system which is basically GNU, but they usually think these are “Linux systems”. At first impression, a “Linux system” sounds like something completely distinct from the “GNU system,” and that is what most users think.

This leads many users to identify themselves as a separate community of “Linux users”, distinct from the GNU user community. They use more than just some GNU programs, they use almost all of the GNU system, but they don’t think of themselves as GNU users. Often they never hear about the GNU idea; if they do, they may not think it relates to them.

Most introductions to the “Linux system” acknowledge that GNU software components play a role in it, but they don’t say that the system as a whole is a modified version of the GNU system that the GNU project has been developing and compiling since Linus Torvalds was in junior high school. They don’t say that the main reason this free operating exists is that the GNU Project worked persistently to achieve its goal of freedom.

As a result, most users don’t know these things. They believe that the “Linux system” was developed by Linus Torvalds “just for fun”, and that their freedom is a matter of good fortune rather than the dedicated pursuit of freedom. This creates a danger that they will leave the survival of free software to fortune as well.

Since human beings tend to correct their first impressions less than called for by additional information they learn later, these users will tend to continue to underestimate their connection to GNU even if they do learn the facts.

When we began trying to support the GNU/Linux system, we found this widespread misinformation led to a practical problem—it hampered cooperation on software maintenance. Normally when users change a GNU program to make it work better on a particular system, they send the change to the maintainer of that program; then they work with the maintainer, explaining the change, arguing for it, and sometimes rewriting it for the sake of the overall coherence and maintainability of the package, to get the patch installed. But people who thought of themselves as “Linux users” showed a tendency to release a forked “Linux-only” version of the GNU program and consider the job done. In some cases we had to redo their work in order to make GNU programs run as released in GNU/Linux systems.

How should the GNU project encourage its users to cooperate? How should we spread the idea that freedom for computer users is important?

We must continue to talk about the freedom to share and change software—and to teach other users to value these freedoms. If we value having a free operating system, it makes sense to think about preserving those freedoms for the long term. If we value having a

variety of free software, it makes sense to think about encouraging others to write free software, instead of proprietary software.

However, it is not enough just to talk about freedom; we must also make sure people know the reasons it is worth listening to what we say.

Long explanations such as our philosophical articles are one way of informing the public, but you may not want to spend so much time on the matter. The most effective way you can help with a small amount of work is simply by using the terms “Linux-based GNU system” or “GNU/Linux system”, instead of “Linux system,” when you write about or mention such a system. Seeing these terms will show many people the reason to pay attention to our philosophical articles.

The system as a whole is more GNU than Linux; the name “GNU/Linux” is fair. When you are choosing the name of a distribution or a user group, a name with “GNU/Linux” will reflect both roots of the combined system, and will bring users into connection with both—including the spirit of freedom and community that is the basis and purpose of GNU.

Index

- +**
- + 28
- +, related option 31
-
- 28
- , related option 31
- blind-mode 32
- blind-mode, related key binding 27
- data-path 33
- display-flip-mode 32
- display-flip-mode, related key binding 29
- environment-mode 32
- environment-mode, related key binding 27
- fullscreen 32
- fullscreen, related key binding 29
- guard-mode 32
- guard-mode, related key binding 27
- help 33
- ignore-environment 31
- ignore-main-config 31
- immortal-mode 31
- immortal-mode, related key binding 27
- inhibit-screensaver 33
- kca 31
- kca, related key binding 28
- kcd 31
- kcd, related key binding 28
- keyboard-flip-mode 33
- keyboard-flip-mode, related key binding 29
- level-module 31
- load-config 31
- load-config, related key binding 29
- mirror-mode 32
- print-paths 33
- skip-title 33
- sound 33
- sound, related key binding 29
- start-level 31
- start-level, related key binding 27
- start-time 31
- start-time, related key binding 28
- time-limit 31
- time-limit, related key binding 28
- total-lives 32
- total-lives, related key binding 27
- usage 33
- version 33
- version, related key binding 29
- video-mode 32
- video-mode, related key binding 27
- window-dimensions 32
- window-position 33
- ? 33
- V 33
- A**
- A 28
- ACTIVATION_POTION 13
- ALT+- 28
- ALT+-, related option 31
- ALT+= 28
- ALT+=, related option 31
- ARCH_BOTTOM 6
- ARCH_TOP_LEFT 10
- ARCH_TOP_MID 10
- ARCH_TOP_RIGHT 10
- ARCH_TOP_SMALL 10
- arches 10
- ASCII 41
- attack movement 25
- B**
- BACK 14
- background, construction 2
- backgrounds 11
- BALCONY 12
- BIG_LIFE_POTION 12
- BIG_PILLAR_BOTTOM 6
- BIG_PILLAR_TOP 6
- BIG_POISON_POTION 12
- BRICKS_00 11
- BRICKS_01 11
- BRICKS_02 11
- BRICKS_03 11
- BROKEN_FLOOR 4
- Bruno Félix Rezende Ribeiro 1
- C**
- C 28
- camera moving 28
- can't fall, con ext 2
- CARPET, ARCH_CARPET_LEFT 10
- CARPET, CARPET_00 9
- CARPET, CARPET_01 9
- carpets 8
- childhood dream 1
- chopper 7
- CHOPPER 8
- climb movement 24
- CLOSER_FLOOR 5
- collidable 4
- command line options 30
- con 2
- con ext can't fall 2

con ext design 3
 con ext event 2
 con ext item 2
 con ext step 2
 configuration 30
 configuration files 30
 configuration options 31
 construction background 2
 construction extension 2
 construction foreground 2, 3
 constructions 2
 conversion, option name 30
 coordinates, level 2
 coordinates, room 28
 copying manual 40
 copleft, FDL and 40
 couch movement, running 22
 counter attack 26
 counter defense 26
 crouch movement mode 20
 crouching jump movement 20
 CTRL+- 28
 CTRL+-, related option 31
 CTRL+= 28
 CTRL+=, related option 31
 CTRL+A 27
 CTRL+G 29
 CTRL+L 29
 CTRL+L, related option 31
 CTRL+Q 27
 CTRL+R 27
 CTRL+S 29
 CTRL+S, related option 33
 CTRL+V 29

D

defense movement 25
 depressible 4
 descend movement 19
 design, con ext 3
 directed relative positioning 15
 DOOR 7
 doors 7
 DOWN 14
 dream childhood 1
 DTD 41
 dynamic 4

E

EMPTY_POTION 12
 environment variables 30
 event, con ext 2
 extension, construction 2

F

F 29
 F, related option 32
 F10 27
 F10, related option 32
 F11 27
 F11, related option 32
 F12 27
 F12, related option 32
 fall movement mode 24
 fall movement, hanging 22
 falling hang movement 24
 falling turn movement 25
 FDL (*see also both manuals and documentation*)
 40, 47
 fight movement mode 25
 fight walk backward movement 25
 fight walk forward movement 25
 files key bindings 29
 FLIP_POTION 13
 FLOAT_POTION 12
 floor 2
 FLOOR 4
 floors 4
 foreground, construction 2, 3
 Free Documentation License (FDL), GNU (*see also*
 FDL, manuals, *and documentation*) ... 40, 47
 FRONT 14

G

game flow key bindings 27
 GNU, GNU Free Documentation License (FDL)
 (*see also FDL, manuals, and documentation*)
 40, 47
 graphical rendering key bindings 27

H

H 28
 hang free 22
 hang movement mode 22
 hang movement, falling 24
 hang non-free 22
 hangable 4
 hangable place 15
 hanging fall movement 22
 hanging turn movement 23
 help options 33
 HIDDEN_FLOOR 6
 HOME 28
 HTML 41

I

I 27
 I, related option 31
 improved 27

item, con ext 2
 items 12

J

J 28
 Jordan Mechner 1
 JPEG 41
 jump movement, crouching 20
 jump movement, normal 18
 jump movement, running 21
 jump movement, vertical 16

K

K 27
 kca 26
 kcd 26
 keep sword movement 26
 key bindings 27
 key bindings, files 29
 key bindings, game flow 27
 key bindings, graphical rendering 27
 key bindings, kid special 28
 key bindings, level navigation 28
 key bindings, life and death 27
 key bindings, miscellaneous 29
 key bindings, peripherals 29
 key bindings, skills 28
 key bindings, time 28
 kid special key bindings 28

L

LATEX 41
 LEFT 14
 level coordinates 2
 level navigation key bindings 28
 level options 31
 LEVEL_DOOR 7
 license documentation 40
 licenses (*see also* Affero, FDL, GPL, LGPL, X11,
 BSD, XFree86, *and* lax permissive licenses)
 47
 life and death key bindings 27
 LOOSE_FLOOR 5

M

manifesto, MININIM 1
 manuals (*see also* manuals, FDL, *and*
 documentation) 40, 47
 Mechner, Jordan 1
 megahit 27
 mirror 8
 MIRROR 8
 miscellaneous key bindings 29
 mode, counter attack 26

mode, counter defense 26
 mode, crouch movement 20
 mode, fall movement 24
 mode, fight movement 25
 mode, hang movement 22
 mode, movement 14
 mode, normal movement 16
 mode, run movement 20
 motivation, MININIM 1
 movement mode 14
 movement mode, crouch 20
 movement mode, fall 24
 movement mode, fight 25
 movement mode, hang 22
 movement mode, normal 16
 movement mode, run 20
 movement, attack 25
 movement, climb 24
 movement, crouching jump 20
 movement, defense 25
 movement, descend 19
 movement, falling hang 24
 movement, falling turn 25
 movement, fight walk backward 25
 movement, fight walk forward 25
 movement, hanging fall 22
 movement, hanging turn 23
 movement, keep sword 26
 movement, normal jump 18
 movement, running crouch 22
 movement, running jump 21
 movement, running turn 22
 movement, turn 16
 movement, vertical jump 16
 movement, walk 18
 movements 14
 moving camera 28

N

N 28
 name, MININIM 1
 NO_BG 11
 NO_BRICKS 11
 NO_FLOOR 4
 NO_ITEM 12
 non-directed relative positioning 3
 normal jump movement 18
 normal movement mode 16

O

oitofelix 1
 OPENER_FLOOR 5
 option name conversion 30
 options, configuration 31
 options, help 33
 options, level 31

options, other	33
options, paths	33
options, rendering	32
options, skills	31
options, time	31
options, window	32
origin, MININIM	1
other options	33

P

paths options	33
PDF	41
peripherals key bindings	29
PILLAR	6
pillars	6
place	2
PNG	41
positioning, directed relative	15
positioning, non-directed relative	3
preface	1

R

R	27
relative positioning, directed	15
relative positioning, non-directed	3
rendering options	32
RIGHT	14
rigid	4
room	2
room coordinates	28
run movement mode	20
running crouch movement	22
running jump movement	21
running turn movement	22

S

screen	2
semi-collidable	4
SGML	41
SHIFT+B	27
SHIFT+B, related option	32
SHIFT+C	28
SHIFT+I	29
SHIFT+I, related option	32
SHIFT+I, related option (mirror mode)	32
SHIFT+K	29
SHIFT+K, related option	33
SHIFT+K, related option (mirror mode)	32
SHIFT+L	27
SHIFT+L, related option	31
SHIFT+S	27
SHIFT+T	27

SHIFT+T, related option	32
SHIFT+W	28
SKELETON_FLOOR	5
skills key bindings	28
skills options	31
SMALL_LIFE_POTION	12
SMALL_POISON_POTION	12
SPACE	28
SPIKES_FLOOR	5
step, con ext.	2
strictly traversable	4
STUCK_FLOOR	5
SWORD	13

T

TAB	28
TCARPET, ARCH_CARPET_LEFT	9
TCARPET, ARCH_CARPET_RIGHT_00	9
TCARPET, ARCH_CARPET_RIGHT_01	9
TCARPET, CARPET_00	8
TCARPET, CARPET_01	9
tile	2
time key bindings	28
time options	31
TORCH	11
trap	4
traversable	4
traversable, strictly	4
turn movement	16
turn movement, falling	25
turn movement, hanging	23
turn movement, running	22

U

U	28
UP	14

V

vertical jump movement	16
------------------------	----

W

walk movement	18
wall	7
WALL	7
WINDOW	11
window options	32

X

XCF	41
XML	41