Package 'replayr'

November 17, 2026

Title Read Input Recordings of Video Games

Version 0.4.0

Description Imports input recordings and demo files of video games, as made in tool-assisted speedrunning, reinforcement learning, and research on human-game interaction. Reads the formats of 'FCEUX', 'BizHawk', 'Dolphin', 'Citra', 'libTAS', 'openMSX', 'JPC-RR', 'Gym Retro', the 'Doom' and 'GoldSrc' engines, and other recording tools. Platforms include the Atari 2600, Nintendo, Sega, and Sony consoles, arcade machines, and personal computers. Imported recordings keep compact digital control arrays, stable control identities, original numeric channels, metadata, and emulator commands. Supports gamepads, keyboards, mice, touchscreens, and recorded motion or light-gun channels. Source timing distinguishes frames, controller polls, and timestamped events. Left and right stick directions, angles, and magnitudes can be selected independently. Allows analysis of input recordings without running an emulator.

License GPL (>= 2)

URL https://github.com/tp451/replayr, https://tp451.github.io/replayr/

BugReports https://github.com/tp451/replayr/issues

Encoding UTF-8

Language en-GB

Depends R (>= 4.0)

Imports jsonlite, xml2, zip (>= 3.0.2), utils

Suggests knitr, rmarkdown, testthat (>= 3.1.7)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Thorben Pelzer [aut, cre] (ORCID:
<https://orcid.org/0000-0001-5039-3779>)

Maintainer Thorben Pelzer <thorben@pelzer.blog>

Repository CRAN

Date/Publication 2026-11-17 12:00:12 UTC

Contents

align_buttons . . . 3
analog_channels . . . 4
as.data.frame.replayr_movie . . . 5
button_layout . . . 6
channel_layout . . . 8
movie_members . . . 9
n_buttons . . . 9
n_observations . . . 10
n_players . . . 11
observation_time . . . 11
observation_unit . . . 12
player . . . 13
polls . . . 14
press_counts . . . 14
read_bk2 . . . 15
read_bk2_chunks . . . 17
read_bkm . . . 19
read_ctas . . . 20
read_ctm . . . 22
read_dem . . . 23
read_dft . . . 25
read_dsm . . . 27
read_dtm . . . 28
read_fbm . . . 31
read_fcm . . . 33
read_fm2 . . . 34
read_fm3 . . . 36
read_fmv . . . 37
read_gbmv . . . 38
read_gmv . . . 40
read_gzm . . . 41
read_jrsr . . . 43
read_lmp . . . 44
read_lsmv . . . 46
read_ltm . . . 48
read_m64 . . . 50
read_mar . . . 51
read_mc2 . . . 53
read_mcm . . . 54
read_mmv . . . 56
read_movie . . . 57

read_omr . . . 59
read_pjm . . . 61
read_pxm . . . 62
read_smv . . . 64
read_tas . . . 65
read_tasproj . . . 67
read_txt . . . 68
read_vbm . . . 70
read_wtf . . . 72
read_ymv . . . 73
read_zmv . . . 75
recording_format . . . 76
recording_info . . . 77
replayr_movie . . . 78
replayr_poll_stream . . . 79
stick_channels . . . 80
subset-recording . . . 81
summary.replayr_movie . . . 82
supported_formats . . . 82
system_channels . . . 83
system_layout . . . 84
universal_buttons . . . 84
universal_button_names . . . 85

Index **86**

`align_buttons` *Align compact controller matrices by stable button identity*

Description

Use this explicitly before comparing different controllers. Missing controls are filled with `FALSE`, which makes an absent control equivalent to an unpressed one for the subsequent analysis. This is a modelling choice. For comparisons of game actions, map the aligned controls to an action vocabulary instead of assuming the same physical button has the same role.

Usage

```
align_buttons(movies, ids = NULL, idx = 1L)
```

Arguments

`movies`	Nonempty list of replayr_movie objects.
`ids`	Numeric or character IDs from `button_layout()` to retain, in output order. Defaults to the union of controls present in the movies, with registered controls first. Controls outside this selection are omitted deliberately.
`idx`	One-based player index, used for every movie.

Value

List of logical matrices with matching columns, named by control key as in `player()`, and stable ID metadata.

Examples

```
paths <- system.file("extdata", c("sample.fm2", "sample.smv"), package = "replayr")
movies <- lapply(paths, read_movie)
aligned <- align_buttons(movies)
vapply(aligned, ncol, integer(1))
```

---

analog_channels *Numeric analogue channels for one player*

Description

Returns the original numeric channels decoded from the recording. These values are retained in every representation mode. No cross-platform normalisation is applied: M64 sticks use signed -128..127 values; PXM/PJM sticks and DTM sticks/triggers use unsigned 0..255 values; BK2 keeps its logged numeric values, including sensors and paddles. Columns are named by key, so that the X axis of the left stick is `left_x` in every format; `channel_layout()` lists the keys beside the labels of the recording. Doom-engine channels are the signed movement and turning commands of each tic, not physical device events. See the format-specific reader for interpretation.

Usage

```
analog_channels(movie, idx = 1L)
```

Arguments

`movie`	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.
`idx`	Player index, 1-based. Defaults to 1 (player 1).

Details

For an unassigned replayr_poll_stream, `idx = 1` selects the single stored stream.

An absent analogue channel has no column. A missing BK2 numeric sample is `NA`, rather than an invented neutral value. Row order matches `player()`.

Value

Numeric matrix with one row per observation and one column per channel, named by the keys of `channel_layout()`; zero columns for a player with no analogue channels.

Examples

```
path <- system.file("extdata", "sample.m64", package = "replayr")
movie <- read_movie(path)
head(analog_channels(movie))
projected <- read_movie(path, analog = "cardinal8")
identical(analog_channels(movie), analog_channels(projected))
```

`as.data.frame.replayr_movie` *A recording as a data frame*

Description

Returns one row per observation of one controller: its position in the file, the button states, the numeric channels, the selected stick representations and the commands. Columns are named by the keys of `button_layout()` and `channel_layout()`. For an unassigned replayr_poll_stream the result is that of `polls()`.

Usage

```
## S3 method for class 'replayr_movie'
as.data.frame(x, row.names = NULL, optional = FALSE, ..., idx = 1L)

## S3 method for class 'replayr_poll_stream'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A replayr_movie or replayr_poll_stream.
`row.names`, `optional`	As in `base::as.data.frame()`; `optional` is not used.
`...`	Not used.
`idx`	Player index, 1-based.

Value

A data frame. `observation` is the position of the row in the file, which differs from the row number after a selection with `[`. A numeric channel whose key is also the key of a button has `_value` appended to its name.

Examples

```
movie <- read_movie(system.file("extdata", "sample.dtm", package = "replayr"))
head(as.data.frame(movie))
```

`button_layout` *Stable controller identities and compact movie columns*

Description

Without a movie, returns the fixed gamepad and peripheral registry. With a movie, returns only its controller's columns, in matrix order. `column` is a local matrix index; `id` and `key` identify the same control across recordings. Unpressed controls supported by the controller are retained. Matrices omit registry entries outside that controller's layout. The registry contains 53 consecutive IDs. Keypad digits and equally numbered Power Pad sensors share `numbered_0` through `numbered_12` (IDs 40-52), with kind `numbered`. Microphone is ID 53. Serial positions in a BK2 LogKey are decoded independently of those sensor numbers.

Usage

```
button_layout(movie = NULL)
```

Arguments

`movie`	An imported recording, or `NULL` for the fixed registry.

Details

Additional named controls, such as keyboard keys, have character IDs of the form `control:<key>`. These are stable within their source key namespace. They appear in the movie's layout without expanding unrelated controllers.

Registered controls use fixed numeric IDs; a mixed layout represents all IDs as character strings. Use keys or IDs for matching, rather than coercing the complete ID column to integers. Source keyboard code sets remain distinct.

Value

Data frame with column, id, key, mnemonic, name and kind.

Which controls share an ID

The same rules apply in every reader, so a console has the same IDs whichever emulator recorded it. `name` shows the label on the device; `key` and `id` are the shared identity. A column that holds several controls of a recording shows their labels joined by a slash, and no two columns of a layout share a name.

- **By label.** A button that carries one of the registry's letters or words (A, B, C, X, Y, Z, L, R, Start, Select, Mode) has that ID. The ID follows the letter, not the place on the pad: A is the right-hand button of a Nintendo pad, the left one of a Genesis pad, and the lower one of an Xbox-style pad, which is what a libTAS recording of an SDL game controller holds.
- **By place, for PlayStation pads.** Their buttons carry shapes. They have the IDs of the SNES buttons in the same places: Cross is B, Circle is A, Triangle is X and Square is Y.

- **In order, for numbered buttons.** The first is A, then B, X, Y, L1, R1, L2 and R2. This covers arcade and joystick buttons, Master System 1 and 2, and PC Engine I to VI. The Neo Geo's row of A, B, C and D is read the same way, so its C and D are X and Y. A device's only fire button or trigger is A.
- **By role, for the rest.** The button that starts or pauses play is Start (PC Engine Run, the Lynx's Pause, the Neo Geo Pocket's Option, keypad Enter, Wii Plus); keypad Clear, Wii Minus, the Lynx's Option 1 and an arcade View Change are Select; Wii Home, libTAS Guide, the Lynx's Option 2 and the PC Engine's switch between two and six buttons are Mode (19); Wii 1 and 2 are X and Y. Mouse left, right and middle are A, B and X, and buttons 4 and 5 are L1 and R1. A light-gun trigger is A; GunCon side A, side B and offscreen shot are B, X and Y.

Arcade buttons that BizHawk shows under the name a cabinet gives them follow one table for the whole platform: Attack and Punch are A, Jump and Missile are B, Kick is X.

Of two direction pads, the left one is the direction pad (IDs 1-4) and the right one has the IDs of the N64's C-buttons (20-23): the Virtual Boy's pads and the WonderSwan's X and Y crosses. An arcade cabinet's two joysticks have the direction IDs of the left and the right analogue stick (N, E, S and W of 24-31 and of 32-39). Controls without a counterpart keep IDs of their own; these are keyboard keys, in the code set of their source. The Intellivision disc has the direction IDs of the left stick: one for each of its eight main directions, and the two neighbours together for a direction between them, such as N and NE for NNE. A disc that a core logs as two axes has the directions the console reads from them. Keypad Star and Pound are Select and Start, like Clear and Enter in the same places. On the Channel F stick, forward and back are Up and Down, the twist is L1 and R1, pushing down is A and pulling up is B. A mouse wheel turned up or down is C-Up or C-Down. A touch on a screen or pad is L3, the press of a pointing input. Each device of an openMSX recording is a controller of its own, with the IDs of its kind. Buttons on the console itself, such as Reset, the Master System's Pause or the lid of a DS, are not part of a pad; they are returned by `system_channels()`.

A shared ID says that two buttons are the same control, not that they do the same thing: what a button does depends on the game. Some recordings store what the game was asked to do, not which button was pressed: a Doom demo holds attack, use and jump, whatever keys or buttons the player had bound to them. These game actions are kept apart from device buttons. They have kind `action` and IDs of the form `action:<name>`, shared by every game that records the same action.

Several source controls under one ID

When the rules place several source controls of one recording under one ID, `source_inputs` retains the decoded input array and `source_layout` identifies its columns and their `universal_key` mapping. That column is active whenever any of them is pressed. Reading with `aliases = FALSE` keeps every control of the recording in a column of its own instead: a control that would be placed under another control's ID keeps the key of its source and an ID of the form `control:<key>`.

Examples

```
path <- system.file("extdata", "sample.m64", package = "replayr")
button_layout(read_movie(path))
```

channel_layout *Names of the numeric channels of one controller*

Description

`analog_channels()` names its columns by key, as `player()` does for buttons. A measurement that several formats record has one key in all of them, whatever the format calls it:

Usage

```
channel_layout(movie, idx = 1L)
```

Arguments

`movie`	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.
`idx`	Player index, 1-based. Defaults to 1 (player 1).

Details

- `left_x`, `left_y`, `right_x`, `right_y`: the axes of the analogue sticks. The N64 stick, the 3DS Circle Pad and the Nunchuk stick are the left stick; the 3DS C-stick is the right one.
- `left_trigger`, `right_trigger`: analogue triggers.
- `mouse_dx`, `mouse_dy`: movement of a mouse since the previous observation. `mouse_x` and `mouse_y` are positions.
- `touch_x`, `touch_y`: position on a touch surface.
- `aim_x`, `aim_y`: where a light gun points.
- `accel_x`, `accel_y`, `accel_z` and `gyro_x`, `gyro_y`, `gyro_z`: motion sensors.

Every other channel has the key formed from its own name in lower case, such as `tilt_x` for the BizHawk field `Tilt X`. Where two channels of a controller would take one shared key, each keeps the key of its own name. Only the names are shared: values keep the range and unit of their source (see `analog_channels()`).

Value

Data frame with one row per channel: `column`, `key`, and `name`, the label of the field in the recording or, for formats without labels, a description of the field.

See Also

`button_layout()` for the digital controls.

Examples

```
path <- system.file("extdata", "paper", "mouse-drag.bk2", package = "replayr")
channel_layout(read_movie(path))
```

movie_members *List input recordings in a ZIP archive*

Description

Some published downloads contain several recordings. Inspect their member names, then pass one as `member` to `read_movie()`. ZIP-based emulator formats such as BK2 use their normal extension and format-specific reader.

Usage

```
movie_members(path)
```

Arguments

`path`	ZIP archive path.

Value

Data frame with member, format, and uncompressed byte count.

Examples

```
# A download that holds two recordings.
archive <- tempfile(fileext = ".zip")
files <- system.file("extdata", c("sample.fm2", "sample.smv"), package = "replayr")
zip::zip(archive, files, mode = "cherry-pick")
movie_members(archive)
read_movie(archive, member = "sample.smv")
unlink(archive)
```

n_buttons *Number of compact controller columns*

Description

Use `button_layout()` to map these columns to stable controller IDs.

Usage

```
n_buttons(movie)
```

Arguments

`movie`	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.

Value

Integer scalar.

Examples

```
path <- system.file("extdata", "sample.m64", package = "replayr")
n_buttons(read_movie(path))
# Stick directions selected with `analog` are further columns.
n_buttons(read_movie(path, analog = "cardinal8"))
```

`n_observations` *Number of observations in an imported recording*

Description

Rows represent the source format's observations: video frames, controller polls, timestamped input events, or game tics; `observation_unit()` names the unit of a recording. With an explicit multi-controller polling order, each row is one cycle through that order. Header video-frame counts remain in the recording metadata.

Usage

```
n_observations(movie)
```

Arguments

`movie`	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.

Value

Integer scalar: the first dimension of `movie$inputs`.

Examples

```
frames <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
n_observations(frames)
observation_unit(frames)
# The rows of an M64 file are controller polls; its header counts the frames.
polls <- read_movie(system.file("extdata", "sample.m64", package = "replayr"))
n_observations(polls)
observation_unit(polls)
```

n_players *Number of assigned input streams*

Description

Keyboard and mouse controls can share a stream. The format-specific header describes device grouping; a stream is not evidence of a distinct person.

Usage

```
n_players(movie)
```

Arguments

movie	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.

Value

Integer scalar: the second dimension of `movie$inputs`. An unassigned poll stream returns `NA_integer_` because the number of participating players cannot be determined from its records.

Examples

```
movie <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
n_players(movie)
# Polls of two controllers without a known order have no number of players.
stream <- read_movie(system.file("extdata", "paper", "unassigned.dtm", package = "replayr"))
n_players(stream)
```

observation_time *Time of each observation*

Description

Returns the time, in seconds from the start of the recording, at which each observation begins. A recording with its own times is read from them: libTAS stores the length of every frame, and JPC-RR and openMSX store a timestamp with every event. A recording of frames or game tics without times is placed on the frame rate of `recording_info()`, where it has one. Observations that are controller polls or game updates have no time in the file, and their number per frame is not fixed; they return `NA`.

Usage

```
observation_time(movie, fallback_rate = NA)
```

Arguments

movie	A replayr_movie or replayr_poll_stream returned by read_movie() or a format-specific reader.
fallback_rate	Frame rate, in frames per second, to use where neither the recording nor its platform gives one, such as 60 or 50 for a Nintendo 64 recording. The default NA assumes none.

Details

After a selection with [, times still count from the start of the file. A frame rate that recording_info() reports as nominal for the platform is that of the original console; an emulator that ran at another speed, or a game that lags, does not change the times returned here.

Value

Numeric vector with one entry per observation, in seconds; NA where the recording fixes no time.

See Also

recording_info(), observation_unit()

Examples

```
movie <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
range(observation_time(movie))
# Seconds for which A is held.
sum(diff(observation_time(movie))[player(movie)[-1, "a"]])
```

---

observation_unit *Unit of observation of an imported recording*

Description

Names what one row of a recording stands for. Most formats store one row per video frame. Others store controller polls, timestamped events, game tics, or several input samples within a frame. n_observations() counts rows whatever their unit, so durations and rates also need this unit and the timing information of the recording.

Usage

```
observation_unit(movie)
```

Arguments

movie	A replayr_movie or replayr_poll_stream returned by read_movie() or a format-specific reader.

Value

A character string: "frame", "scripted frame", "input subframe", "controller poll", "controller polling cycle", "Wii Remote poll", "device poll", "event", "game tic" or "game update".

Examples

```
frames <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
observation_unit(frames)
polls <- read_movie(system.file("extdata", "sample.m64", package = "replayr"))
observation_unit(polls)
```

---

player *Button states for one controller*

Description

Button states for one controller

Usage

```
player(movie, idx = 1L)
```

Arguments

movie	A replayr_movie or replayr_poll_stream returned by read_movie() or a format-specific reader.
idx	Player index, 1-based. Defaults to 1 (player 1).

Value

Logical matrix, n_observations(movie) rows by n_buttons(movie) columns. Columns are named by the control keys of button_layout(), as in press_counts(), align_buttons() and polls(), so that player(movie)[, "a"] is the A button in every recording. The attributes button_ids and button_names hold the IDs and the labels of the columns; button_layout() has the whole layout.

Examples

```
path <- system.file("extdata", "sample.fm2", package = "replayr")
mv <- read_movie(path)
p1 <- player(mv)
colSums(p1)        # number of frames each button was held
```

polls *Extract an unassigned controller-poll stream*

Description

Extract an unassigned controller-poll stream

Usage

```
polls(x)
```

Arguments

x	A replayr_poll_stream.

Value

Data frame in source order. `poll` is the position of the poll in the file. Digital and numeric columns are named by the keys of `button_layout()` and `channel_layout()`. For M64, `commands` uses bit 0 for reset. For DTM, bit 0 is disc change and bit 1 is reset. Standard DTM exposes `is_connected` and `get_origin`; Zelda Edition exposes `loading`, `tuner_event`, `link_x`, and `link_z` (see `read_dtm()`). The object's `poll_bytes` component contains four original bytes per M64 poll, eight per standard DTM poll, or seventeen per Zelda Edition poll, in source order. CTM stores seven bytes per device event. Wii stores padded report bytes with `poll_lengths`; mixed HID records include the length byte. `source_tail` retains incomplete final Wii records. Use observation flags before analysing device-specific fields.

Examples

```
stream <- read_movie(system.file("extdata", "paper", "unassigned.dtm", package = "replayr"))
stream
polls(stream)[, c("poll", "port", "a", "left_x", "left_y", "is_connected")]
# A selection keeps the positions of the polls in the file.
polls(stream[2:3])$poll
```

press_counts *Held-observation counts per control*

Description

Held-observation counts per control

Usage

```
press_counts(movie, idx = 1L)
```

Arguments

movie	A replayr_movie or replayr_poll_stream returned by read_movie() or a format-specific reader.
idx	Player index, 1-based.

Value

Named integer vector of length n_buttons(movie). Names are the control keys of button_layout(), as for the columns of player(). Counts give the observations held, rather than the number of off-to-on transitions.

See Also

summary() for the number of activations.

Examples

```
movie <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
press_counts(movie)
# As a share of the recording, for the second controller.
round(press_counts(movie, 2L) / n_observations(movie), 3)
```

---

read_bk2 *Read a BK2 recording*

Description

BizHawk ZIP movies. LogKey names determine digital controls and numeric channels. Known N64 and PlayStation stick encodings can be transformed; other numeric channels remain raw. The BK2 files that Gym Retro writes are read as well. Their input log names the controls without the LogKey: prefix and its Reset and Power group; the reader restores both and notes it in movie$header$logKeyRestored. Buttons 1 and 2 of a Master System or Game Gear pad, which Gym Retro writes as B and A, are named B1 and B2 as in a BizHawk recording. An arcade cabinet's analogue joysticks are its left and right analogue stick. MAME records their raw values without centre or range, which differ from cabinet to cabinet; when directions or angles are requested, both are estimated from the values of the recording, with a warning that states the estimate. Digital columns are compact; button_layout() gives their stable IDs. Numeric source channels are always retained in analog_channels(). The full import defaults to a 256 MiB limit on the expanded input log to avoid large temporary allocations. read_bk2_chunks() processes larger recordings in bounded memory. The final movie and parsing buffers can use substantially more memory than the text itself.

Usage

```
read_bk2(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  max_input_bytes = 256 * 1024^2,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
max_input_bytes	Maximum expanded input-log size for a full import, in bytes. The default is 256 * 1024^2. Increase this explicitly when sufficient memory is available, up to R's single-string limit of 2147483647 bytes. The limit is checked before extracting the input log.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

Like BizHawk, the reader takes input rows and the LogKey only from lines that start with them. If the LogKey line is indented, its control layout is still used and a warning is given; indented input rows stay excluded and are counted in `header$indentedInputLines`. A log without a LogKey is read only when its header and every row match the standard two-pad NES layout; any other log without a LogKey raises an error. A digital control is held unless its character is `.`, as BizHawk reads it; the letter written there is a convention.

BizHawk writes the hash of the game under `header$SHA1`, but several cores write an MD5 or a CRC32 there. `header$SHA1Kind` says which it is: `"SHA1"`, `"MD5"` or `"CRC32"`, and `""` for a value that is no hash, such as `"N/A"`. The MD5 and CRC32 of a disc are BizHawk's hashes of part of the disc, not those of an image file.

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_bk2(system.file("extdata", "sample.bk2", package = "replayr"))
movie
head(system_channels(movie), 3)
# A mouse recorded by BizHawk: its buttons are controls, its movement
# is two numeric channels.
mouse <- read_bk2(system.file("extdata", "paper", "mouse-drag.bk2", package = "replayr"))
button_layout(mouse)[, c("key", "name")]
channel_layout(mouse)
```

---

read_bk2_chunks *Process a BizHawk recording in bounded chunks*

Description

The input member is decompressed once to a temporary file and its ZIP checksum is verified before any callback runs. The temporary file needs disk space equal to the uncompressed log and is removed on exit. It is read sequentially, keeping memory use bounded. Each callback receives a configured movie containing at most `chunk_size` observations. Its header includes `firstObservation` and `lastObservation`, using one-based positions in the original recording. Return a small summary from `FUN` to keep memory use bounded; returning the movie itself retains every chunk in memory.

Usage

```
read_bk2_chunks(
  path,
  FUN,
  chunk_size = 100000L,
  analog = "raw",
```

```
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
FUN	Function called with one movie chunk. Additional arguments in ... are passed to this function.
chunk_size	Maximum observations passed to each callback.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Additional arguments for FUN.

Value

A list of callback results, with observations and chunks attributes giving the total processed. An empty recording gives a list with no callback results and zero observations.

Examples

```
path <- system.file("extdata", "expanded", "two-stick-3ds.bk2", package = "replayr")
```

```
counts <- read_bk2_chunks(path, function(chunk) n_observations(chunk), chunk_size = 2L)
sum(unlist(counts))
attr(counts, "observations")
```

---

read_bkm *Read a BKM recording*

Description

Legacy BizHawk text recordings. Digital controls have compact columns with stable identities. Numeric source fields remain available through `analog_channels()`. As in `read_bk2()`, `header$SHA1Kind` says whether the hash under `header$SHA1` is a SHA1, an MD5 or a CRC32.

Usage

```
read_bkm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.

`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
`aliases`	`TRUE` (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With `FALSE`, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
`...`	Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Value

An imported recording with header metadata and original input fields.

Examples

```
movie <- read_bkm(system.file("extdata", "expanded", "legacy-n64.bkm", package = "replayr"))
movie
analog_channels(movie)
```

`read_ctas` *Read a CoffeeTools recording*

Description

Reads CTAS versions 3 and 4 of CoffeeTools, a recording tool for the game *UFO 50*. Each row contains one frame's input for both players: four directions, `ufo50_fire1`, `ufo50_fire2`, and Start. The Start column records a press event; the other six columns record held states.

Usage

```
read_ctas(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields

through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.

`deadzone` Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.

`triggers` `"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

`aliases` `TRUE` (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With `FALSE`, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

`movie$pressed` and `movie$released` retain the six controls' separate press and release flags as frame-by-player-by-control logical arrays. For example, `movie$pressed[, 1, "ufo50_fire1"]` selects player 1's first action-button press events. These flags are read from the file alongside held states, allowing a game's press and hold behaviour to be compared.

`movie$randomization` contains the recorded `input_frame` and `seed` pairs in source units. `movie$source_words` retains both players' input words, and `movie$source_header` retains the original header. Version 4 adds `rerecord_count` and `current_time` to `movie$header`; those fields are `NA` for version 3. Rows use the file's frame index; randomisation frame numbers are retained exactly as stored.

Value

A replayr_movie with held states, press/release flags, and randomisation data.

Examples

```
path <- system.file("extdata", "computers", "two-player-actions.ctas",
  package = "replayr")
movie <- read_ctas(path)
movie
as.data.frame(movie, idx = 2L)
```

read_ctm *Read a Citra device-poll recording*

Description

Imports Nintendo 3DS CTM recordings as ordered device polls. Pad and Circle Pad, touch, accelerometer, gyroscope, IR C-stick, and extra-HID polls each occupy their own rows. polls() exposes event_type and flags identifying the observed device. observed_inputs marks the digital controls measured at each poll; numeric fields for other devices are NA.

Usage

```
read_ctm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

analog_channels() retains the source coordinates, motion values, and battery level. The Circle Pad and C-stick support the four stick modes through analog, using their source-specific centres and scales. poll_bytes retains the seven-byte records. Timing follows device-poll order; CTM provides no video-frame assignment for these rows.

Value

A replayr_poll_stream containing ordered device polls and metadata.

Examples

```
stream <- read_ctm(system.file("extdata", "legacy", "devices.ctm", package = "replayr"))
# Each row is the poll of one device: pad, touch screen or a sensor.
class(stream)
observation_unit(stream)
channel_layout(stream)$key
```

---

read_dem *Read a GoldSrc demo*

Description

Reads the demos of the Half-Life engine (GoldSrc): *Half-Life*, *Counter-Strike* 1.x and the other games built on it, demo protocol 5, from the engine versions of 2001 (network protocols 42 and 43) to the current one (48). A demo stores, with every network message the client received, the command the client was sending at that moment. Each such frame is one observation.

Usage

```
read_dem(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

player() returns the 16 actions of the command. They are what the game was asked to do, not keys: which key or mouse button is bound to an action is a setting of the player. The actions have action: identities of their own, shared with other games that record the same action; attack, use and jump are those of Doom-engine demos.

analog_channels() returns the movement the command asks for (forwardmove, sidemove, upmove), the view angles it asks for in degrees (view_pitch, view_yaw, view_roll), the impulse number, the selected weapon and the duration of the command in milliseconds (msec). The view angles are the player's aim as the client sends it: their change from one observation to the next is the turning asked for with the mouse or the turn keys. The game can also set them, for example when the player respawns.

movie$timing gives for each observation its section, the seconds and the frame number the demo stores (both count anew in every section), and whether it is a start frame of the loading section. Two network messages can arrive in one frame; they are two observations with one frame number. observation_time() runs on from section to section. movie$sections lists the sections of the

demo, and `movie$console_commands` the console commands it recorded, among them the key actions (`+jump`, `-jump`).

A demo recorded by a broadcast proxy (HLTV) holds no commands of a player: every command in it is empty. The reader says so in a warning and in `movie$header$playerCommands`. A demo that was not closed or was cut short is read up to its last complete frame, with a warning. Demos of demo protocol 4, which the engines of early 2001 wrote, store no command with their frames and are refused. Quake and Source engine demos share the extension `.dem` and are other formats.

Value

A replayr_movie with one player.

Examples

```
path <- system.file("extdata", "computers", "goldsrc-moves.dem", package = "replayr")
movie <- read_dem(path)
colSums(player(movie))
head(analog_channels(movie)[, c("forwardmove", "view_yaw")])
```

---

read_dft *Read a Dustmod input script archive*

Description

Imports the DFT archives of Dustmod, a modified version of the game *Dustforce DX*, by following their entry script and INCLUDE commands. Each row represents one scripted input frame. Directional double taps are preserved in `double_tapped`; `source_digits` and `source_values` retain the original 0, 1, and 2 encoding. Optional taunt and restart columns are retained when present.

Usage

```
read_dft(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  entry = NULL,
  ...
)
```

Arguments

`path`	Recording file path.

analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
entry	Entry script member. NULL selects the archive's unique main.txt; an explicit member resolves archives with several candidates.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

source_locations identifies the archive member and line for each row. source_commands preserves loading, synchronisation, mouse, statistics, and include commands in script order. MOUSE coordinates and button states apply to the next scripted row, with observation masks identifying those rows. SYNCLOAD can add runtime waiting whose duration requires execution in the game; the imported rows retain script order.

Value

A replayr_movie with source scripts, commands, and locations.

Examples

```
movie <- read_dft(system.file("extdata", "computers", "menu-double-tap.dft", package = "replayr"))
# Jump, dash and the attacks are actions of the game, not buttons.
button_layout(movie)[, c("id", "key", "kind")]
head(analog_channels(movie), 3)
```

`read_dsm` *Read a DeSmuME recording*

Description

Imports Nintendo DS DSM versions 1 and 2 in text or binary form, with one row per recorded frame. `player()` includes the pad buttons, touch state, microphone activation, and lid state. `button_layout()` identifies the fields.

Usage

```
read_dsm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

analog_channels() returns touch_x, touch_y, and touch_pressed. Touch coordinates use the screen's top-left origin, with a nominal extent of 256 by 192. Text recordings with a fourth numeric field also provide microphone_sample; absent samples are NA. The binary layout stores three touch fields. Stick options leave these source channels unchanged.

The commands component retains the original byte: microphone activation uses bit 0, reset uses bit 1, and lid state uses bit 2. Text records may be adjacent or separated by whitespace.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_dsm(system.file("extdata", "legacy", "touch-v2.dsm", package = "replayr"))
movie
analog_channels(movie)
```

---

read_dtm *Read a DTM recording*

Description

Dolphin GameCube and Wii recordings. Multiple enabled ports return an ordered replayr_poll_stream with unassigned ports. An optional controller_order constructs player streams using a caller-verified repeating port order.

Usage

```
read_dtm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  controller_order = NULL,
  dtm_variant = "auto",
  wii_variant = "auto",
  wii_extension = "raw",
```

```
  wii_key = NULL,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
controller_order	Optional integer sequence listing every enabled controller port once, in its verified repeating polling order. Digital columns are compact; button_layout() gives their stable IDs. Numeric source channels are always retained in analog_channels().
dtm_variant	"auto" (default), "standard" (8-byte records), or "zelda" (17-byte Dolphin Zelda Edition records). Automatic recognition uses the verified Zelda Edition revision a6e66c227c592d09ff09ef9b238d15ff52dd9632; other revisions default to the standard layout. Select "zelda" explicitly for another build whose extended layout has been verified from its source.
wii_variant	"auto", "hid", "hid-prefix8", "hid-nomarker", "serialized", "mixed-hid", "mixed-hid-nomarker", or "mixed-hid-connected". Automatic selection uses packet structure and, for mixed recordings, verified Dolphin revision profiles. "hid-nomarker" handles reports starting with the Wii report ID. "mixed-hid-nomarker" implements the native encoder rule: connected GameCube reports set the connection flag, disconnected reports have zero controls (with optional disc/reset

commands), and Wii reports carry a prescribed length and report ID. Noncanonical records are rejected. Profiles for Dolphin 5.0-20571 and 2407 are selected automatically from the stored revision. `header$framingRule` records the encoding rule. `"mixed-hid-connected"` remains available for externally verified files requiring the additional assumption that every GameCube poll is connected; that assumption is recorded in `header$framingAssumption`. `polls()` retains device family, connection flags, and source order. Individual ports remain unassigned when several same-family ports are enabled.

`wii_extension` Legacy extension interpretation: `"raw"` (default), `"nunchuk"`, or `"classic"`. A selected type assumes unencrypted reports unless an encryption key is supplied.

`wii_key` Optional 16-byte numeric vector: the eight `ft` bytes followed by the eight `sb` bytes of the verified Dolphin extension key.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

Every complete stored poll is imported. Dolphin's header input statistic is retained as `header$inputCount`, alongside `header$availablePollCount` and their signed difference in `header$pollCountDifference`. Dolphin uses the input count as a statistic, so differences remain metadata. `header$frameCount` retains the separate video-frame count. `header$romId` is the game ID of the disc, such as `"GALE01"`, and `header$romMD5` the MD5 of the disc image, where Dolphin computed one.

Zelda Edition extends each poll with a loading flag, a Tingle Tuner event code, and Link's X/Z coordinates used by the emulator to detect desyncs. These are available in `poll_fields`, with one row per original poll, including when `controller_order` groups polls into player cycles. Columns are `loading`, `tuner_event`, `link_x`, and `link_z`; coordinates retain the game's source units. `poll_bytes` retains all original bytes. Wii Remote recordings retain buttons, source motion values, and IR coordinates. Wii Remote 1/2 use X/Y (IDs 9/10), +/- use Start/Select (5/6), and Home uses Mode (19). Its D-pad and A/B buttons share the gamepad IDs. Nunchuk C/Z use 17/18, Classic Controller buttons use the corresponding gamepad IDs, and libTAS Guide uses Mode. Light-gun triggers use A (8); GunCon side A/B and offscreen shot use B/X/Y (7/9/10). Mouse left, right, and middle buttons use A/B/X. See `button_layout()` for shared identities.

When multiple source buttons share an ID, the column is active if either is pressed. `source_inputs` and `source_layout` preserve the separate states and their mapping whenever aliases are combined. Wii device states also appear in `poll_fields` as `wii_remote_*` and `wii_classic_*` columns, which `polls()` includes for unassigned streams. Legacy HID reports retain extension bytes; `wii_extension = "nunchuk"` or `"classic"` decodes a caller-identified extension. Supply `wii_key` when its reports are encrypted. The default retains the bytes because older DTM files do not store an extension identity or encryption key. New serialised Dolphin states include their extension type; Nunchuk and Classic Controller states are decoded automatically. Other extension payloads remain source bytes. Each row is a poll. `poll_lengths` gives valid bytes in the padded `poll_bytes` matrix; historical eight-byte prefixes remain in `poll_prefix_bytes`. Wii Remote acceleration is given on Dolphin's ten-bit scale. A report stores nine bits for Y and Z, which are widened as Dolphin widens them, by repeating the lowest stored bit. Mixed GameCube/Wii streams from the verified legacy Dolphin layout retain device family and physical port (when unique) on each poll. They stay in source order and do not accept `controller_order`. Automatic mixed recognition uses revision `7804f388aeb075266ae8c363e0932c355ba6ed5e`; select `mixed-hid` explicitly only

for a build with the same verified record layout. An incomplete final Wii packet is retained in `source_tail`, with a warning. Invalid reports retain their raw bytes and have `valid_report = FALSE`. `poll_fields$report_issue` describes the problem, and `source_offset` gives the zero-based byte position in the file. Filter digital fields using the validity flag before analysis.

Value

A replayr_movie for assigned ports, or a replayr_poll_stream for a multi-controller recording with unassigned ports.

Examples

```
movie <- read_dtm(system.file("extdata", "sample.dtm", package = "replayr"))
head(analog_channels(movie), 3)
# Polls of several controllers without a known order stay one stream.
stream <- read_dtm(system.file("extdata", "paper", "unassigned.dtm", package = "replayr"))
class(stream)
polls(stream)[, c("poll", "port", "a", "left_x", "left_y")]
```

---

read_fbm *Read a FinalBurn Alpha recording*

Description

Imports FinalBurn Alpha FBM input streams using an explicit game input layout. FBM stores values in driver order, so the recording must be paired with a verified ROM identifier or an ordered schema. Bundled game layouts cite their driver sources in `extdata/fbm-layouts.json`.

Usage

```
read_fbm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  game = NULL,
  input_schema = NULL,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields

through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.

deadzone — Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

triggers — "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases — TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

game — A verified ROM identifier matching a bundled input layout, such as "ghouls". Supply either game or input_schema.

input_schema — An ordered schema with a source reference and a fields list. Each field gives key, player (zero for system inputs), type ("button", "byte", or "analog"), and an optional display name. Optional button_encoding is "binary" (default), "bit0", or "nonzero", according to the driver's interpretation of the byte. Field order must match the emulator driver exactly.

... — Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

player() returns game controls, analog_channels() retains numeric player inputs, and system_channels() returns cabinet and system inputs. It also retains every raw field under raw_pN_key or raw_system_key, including source bytes whose button interpretation uses only bit zero. Rows follow the stored frame order. The header retains the frame count, rerecord count, source flags, and input-schema reference, and in metadata the text that the author of the recording entered when recording, where the file has one (usually the author's name or a comment).

Value

A replayr_movie with compact controls and the input schema.

Examples

```
path <- system.file("extdata", "expanded", "synthetic-arcade.fbm",
```

```
                    package = "replayr")
movie <- read_fbm(path, game = "ghouls")
press_counts(movie)
```

`read_fcm` *Read a FCM recording*

Description

FCEU binary NES movies, including supported delta-stream versions. Digital columns are compact; `button_layout()` gives their stable IDs. Numeric source channels are always retained in `analog_channels()`.

Usage

```
read_fcm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.

triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

commands holds one bit for every control code applied in a frame, bit code - 1: 1 is a reset, 2 a power cycle, 64 a VS System coin, 8388608 an FDS disk insert, 16777216 an FDS eject and 33554432 an FDS side select.

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_fcm(system.file("extdata", "sample.fcm", package = "replayr"))
movie
n_players(movie)
```

---

read_fm2 *Read a FM2 recording*

Description

FCEUX text movies; NES and Famicom button states. Digital columns are compact; button_layout() gives their stable IDs. Numeric source channels are always retained in analog_channels().

Usage

```
read_fm2(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

A position of a gamepad field is held unless it shows . or a space, as FCEUX reads it; the letter written there is a convention. Controllers follow the port declarations: a port declared empty is no controller, and header$controllerPorts gives the port of each player. As in FCEUX, a file is an FM2 movie only if it begins with version 3; any other file is refused.

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
path <- system.file("extdata", "sample.fm2", package = "replayr")
movie <- read_fm2(path)
movie
recording_info(movie)
```

`read_fm3` *Read a FM3 recording*

Description

FCEUX TAS Editor projects, with the movie length separating input records from editor data.

Usage

```
read_fm3(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
`aliases`	`TRUE` (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With `FALSE`, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

... Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

An imported recording containing original input fields and metadata.

Examples

```
movie <- read_fm3(system.file("extdata", "expanded", "movement-jumps.fm3", package = "replayr"))
summary(movie)
```

---

read_fmv *Read a Famtasia recording*

Description

Imports NES and Famicom Disk System input, with one row per recorded frame and one player for each enabled controller port. player() returns the pad buttons, and button_layout() identifies their columns.

Usage

```
read_fmv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's

controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

`triggers` "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

`aliases` TRUE (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

The `commands` component retains the Famicom Disk System command byte when present. `source_packets` retains every complete packet. An incomplete final packet produces a warning and remains available in `source_tail`. The header preserves enabled ports, rerecord count, and movie metadata.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_fmv(system.file("extdata", "legacy", "buttons.fmv", package = "replayr"))
movie
press_counts(movie)
```

---

read_gbmv *Read a GBMV recording*

Description

GBAHawk ZIP recordings. Digital controls have compact columns with stable identities. Numeric source fields remain available through `analog_channels()`.

Usage

```
read_gbmv(
  path,
  analog = "raw",
  deadzone = NULL,
```

```
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

An imported recording with header metadata and original input fields.

Examples

```
movie <- read_gbmv(system.file("extdata", "expanded", "gba-input.gbmv", package = "replayr"))
movie
as.data.frame(movie)
```

read_gmv *Read a GMV recording*

Description

Gens recordings with three- or six-button controllers. Controller capability comes from the header, with older unspecified types retained explicitly. Digital columns are compact; `button_layout()` gives their stable IDs. Numeric source channels are always retained in `analog_channels()`.

Usage

```
read_gmv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

A two-controller recording stores X, Y, Z and Mode for both controllers whatever their declared type. For a controller declared as three-button these four lie outside its layout; commands retains them as pressed bits, 1, 2, 4 and 8 for controller 1 and 16, 32, 64 and 128 for controller 2.

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_gmv(system.file("extdata", "sample.gmv", package = "replayr"))
movie
button_layout(movie)[, c("id", "key", "name")]
```

---

read_gzm *Read a gz input macro*

Description

Imports the macros of gz, a practice version of the Nintendo 64 game *The Legend of Zelda: Ocarina of Time*, with one row per recorded game update. player() returns the N64 buttons. analog_channels() retains signed left_x and left_y; analog selects a derived stick representation.

Usage

```
read_gzm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

frame_fields retains pad deltas and press/release masks. system_events retains random-seed changes and any ocarina-input, audio-synchronisation, or room-load records. The commands component marks reset events, and source_packets retains the six-byte input records. Game updates provide the observation order; elapsed time requires the game's update timing.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_gzm(system.file("extdata", "legacy", "buttons.gzm", package = "replayr"))
observation_unit(movie)
head(analog_channels(movie), 3)
```

read_jrsr *Read a JPC-RR recording*

Description

Reads JRSR text archives containing an `events` section. Keyboard edges, mouse buttons and movement, joystick events, and reset commands are decoded. Other emulator events remain in `movie$events` with their arguments and original source lines. Initialisation sections remain in `movie$header$initialization` and `movie$event_sections`.

Usage

```
read_jrsr(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

Each row follows one timestamped event. movie$events$timestamp_ns gives absolute time in nanoseconds after applying recorded RELATIVE and ABSOLUTE options. timestamp_source retains the source timestamp text. Keyboard and button states persist until the next relevant event; mouse movement channels record the displacement at that event.

Keyboard identities use jpcrr_key_<number>, preserving JPC-RR's key-code namespace. The Pause event appears as a jpcrr_pause pulse. Keyboard and mouse inputs occupy player 1. analog_channels() exposes mouse_dx, mouse_dy, and mouse_wheel, and the joystick channels joystick_x_hold_ns and joystick_y_hold_ns. Joystick values are recorded game-port hold times in nanoseconds and remain NA before their first update.

Value

A replayr_movie with event times, reconstructed states, and source events.

Examples

```
movie <- read_jrsr(system.file("extdata", "computers", "keyboard-mouse.jrsr", package = "replayr"))
observation_unit(movie)
observation_time(movie)
analog_channels(movie)[1:4, ]
```

---

read_lmp *Read a Doom-engine demo*

Description

Imports demos of Doom, Heretic and Hexen and of the source ports that keep their demo format: Doom 1.2 and earlier, Doom 1.4 to 1.9 (versions 104-109), TASDoom (110), Doom 1.91 (111), Boom (200-202), LxDoom and MBF (203), PrBoom (210-214) and MBF21 (221), with or without the outer header that dsda-doom or PrBoom+um put in front. Each row is a game tic at 35 tics per second. header$enabledPorts identifies the players whose commands were stored.

Usage

```
read_lmp(
  path,
  analog = "raw",
  deadzone = NULL,
```

```
  triggers = "raw",
  aliases = TRUE,
  game = NULL,
  longtics = NULL,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
game	"doom", "heretic" or "hexen". Required for a demo without a version byte; every other demo identifies its own layout.
longtics	For Heretic and Hexen demos: TRUE if turning was recorded with 16 bits and the header does not say so, FALSE to rule that out, NULL (default) to follow the header.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

A demo stores what the game was asked to do in each tic, not the keys or buttons behind it. player() returns the actions Attack and Use, and Jump for Hexen and for demos with dsda-doom's header; button_layout() lists them with IDs of their own (action:attack, action:use,

action:jump). analog_channels() returns signed forwardmove, sidemove and angleturn: how fast the player asked to move forward and sideways, and how far to turn in that tic, in 1/256 of a revolution. Demos with 16-bit turning keep the finer steps as fractions of that unit. Heretic and Hexen add look and fly, from -7 to 7; -8 asks to centre the view or to stop flying. All of these are commands assembled from the input devices, so stick options leave them unchanged.

Weapon selection and special-command flags remain in header$weaponselect and header$has_special; commands retains the special-command bytes. Artifact commands of Heretic and Hexen are in header$artifact, Hexen's player classes in header$playerClass, and dsda-doom's extended commands in header$extendedCommands. Game options remain in header$gameOptions.

Demos of Doom 1.2 and earlier, of Heretic and of Hexen begin with the skill level and do not name their game, and the three layouts differ, so game is required for them. Heretic and Hexen demos can store 16-bit turning; later ports mark this in the header, Heretic+ and Hexen+ do not, and longtics = TRUE reads such a demo. A demo that fits the file only when read that way is refused with a message that says so.

source_header and source_packets preserve the recorded bytes, and source_tail preserves bytes after the demo-end marker: c(source_header, t(source_packets), as.raw(0x80), source_tail) is the file.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
demo <- read_lmp(system.file("extdata", "sample.lmp", package = "replayr"))
button_layout(demo)
head(analog_channels(demo))
```

`read_lsmv` *Read an lsnes LSMV recording*

Description

Reads ZIP input logs with controlsversion 0: Game Boy/Game Boy Color pads, SNES 12- and 16-button pads, multitaps, Y-cabled pads, and a declared Super Scope on port 2. Super Game Boy movies retain their SNES controller layout. The four spare bits of a 16-bit pad report are not buttons of the pad; system_channels() returns them as pad<n>_ext0 to pad<n>_ext3.

Usage

```
read_lsmv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  max_input_bytes = 256 * 1024^2,
```

```
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
max_input_bytes	Maximum expanded input-log size for a full import, in bytes. The default is 256 * 1024^2. Increase this explicitly when sufficient memory is available, up to R's single-string limit of 2147483647 bytes. The limit is checked before extracting the input log.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

Every non-empty input row is retained, including subframes between frame starts. n_observations() counts these observations; header$frameCount counts frame starts. subframes contains source_line, frame_start, source_frame, and subframe (the row's one-based position within its source frame). Player indices follow controller fields in the source log.

subframes also retains reset, hard_reset, reset_delay_high, and reset_delay_low. The encoded delay is 10000 * reset_delay_high + reset_delay_low. commands uses bit 0 for the reset

flag and bit 1 for the hard-reset flag. `source_input` preserves the complete input text, including blank lines and original line endings.

Super Scope controls use `a` (trigger), `b` (cursor), `mode` (turbo), and `start` (pause). Their separate source keys remain in `source_layout`. `analog_channels()` returns its signed source coordinates as `aim_x` and `aim_y`. `button_layout()` gives stable control IDs. Mouse and Justifier port types raise an error.

The input member's ZIP checksum is verified before parsing. Full imports default to a 256 MiB expanded-input limit. Damaged controller metadata raises an error; other complete metadata with an integrity defect is retained with a warning.

Value

A replayr_movie with one observation per input subframe.

Examples

```
movie <- read_lsmv(system.file("extdata", "sample.lsmv", package = "replayr"))
observation_unit(movie)
head(movie$subframes[, c("source_frame", "subframe", "reset")], 3)
```

---

read_ltm *Read a libTAS recording*

Description

Reads native LTM archives containing `config.ini` and `inputs`. Supports the positional input format through libTAS 1.3.5 and the later tagged format, including ordered input events and up to four controllers.

Usage

```
read_ltm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields

through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.

deadzone — Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

triggers — "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases — TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

... — Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

Each row is a recorded frame. movie$timing contains framerate_num, framerate_den, duration_seconds, realtime_sec, and realtime_nsec. Frame durations follow the recorded rate changes. movie$input_events retains each frame's ordered events, and movie$flags retains emulator flags.

source_input_log retains the original input lines and their row index. Keyboard keys use their hexadecimal X KeySym in button_layout(), such as keysym_ff53 for Right. Keyboard and mouse inputs occupy player 1 alongside controller 1. Controller buttons retain gamepad identities, including L3 and R3. analog_channels() includes left_x, left_y, right_x, right_y, left_trigger, and right_trigger as signed 16-bit source values. Both sticks support the representations selected by analog. Mouse channels are mouse_x, mouse_y, mouse_wheel, and mouse_mode; mode 0 denotes absolute coordinates and mode 1 denotes relative movement. The complete configuration remains in movie$header$configuration.

Value

A replayr_movie with frame timing, source inputs, and metadata.

Examples

```
movie <- read_ltm(system.file("extdata", "computers", "dual-stick-menu.ltm", package = "replayr"))
movie
channel_layout(movie)
# Every frame has its own length.
observation_time(movie)
```

read_m64 *Read a M64 recording*

Description

Mupen64 version 3 recordings. A single enabled controller returns a replayr_movie. Multiple enabled controllers return a replayr_poll_stream with all samples in source order, because M64 stores no per-sample port IDs. Digital columns are compact; `button_layout()` gives their stable IDs. Numeric source channels are always retained in `analog_channels()`.

Usage

```
read_m64(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  controller_order = NULL,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
triggers	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases — TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

controller_order — Optional sequence of distinct enabled controller ports (1–4) in an externally verified repeating polling order. NULL preserves an unassigned stream when multiple ports are enabled.

... — Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

An externally verified repeating polling order can be supplied to construct per-player matrices. These use compact player indices in ascending physical port order; movie$header$controllerPorts gives the physical port for each player index. Enabled ports omitted from that order are treated as unpolled. Each output row is one supplied polling cycle, not a video frame. The source poll count and VI count remain in header$inputSampleCount and header$viCount; poll_bytes retains four original bytes per source poll. Import stops at the declared sample count, as Mupen playback does. Legacy writers can leave an extra terminal record or an older file suffix; these bytes remain in source_tail.

The header identifies the ROM by its internal name (header$romName), the CRC1 of its header (header$romHeaderCRC, "46039fb4" for the North American *Kirby 64: The Crystal Shards*), its country code (header$romCountry, such as "E" for North America and "J" for Japan) and its version (header$romVersion).

Value

A replayr_movie for an assigned controller order, or a replayr_poll_stream for multiple controllers with unassigned ports.

Examples

```
movie <- read_m64(system.file("extdata", "sample.m64", package = "replayr"))
observation_unit(movie)
head(analog_channels(movie), 3)
# The stick as eight directions beside the buttons.
directions <- read_m64(system.file("extdata", "sample.m64", package = "replayr"),
  analog = "cardinal8")
n_buttons(directions)
```

---

read_mar *Read a MAME recording*

Description

Reads MAME-RR input-port records. Known games use the bundled port layouts; other games require an explicit input schema. Original port words, timestamps, and analogue accumulator fields remain available in system_channels.

Usage

```
read_mar(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  input_schema = NULL,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
input_schema	A game-specific port schema; see the bundled extdata/mar-layouts.json for the schema structure.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

An imported recording with source port values and metadata.

Examples

```
path <- system.file("extdata", "expanded", "arcade-dual-joysticks.mar",
  package = "replayr")
movie <- read_mar(path)
# The two joysticks of a cabinet are the left and the right stick.
button_layout(movie)[, c("key", "name")]
names(system_channels(movie))
```

`read_mc2` *Read a PCEjin or VBjin recording*

Description

Imports version-1 text and binary recordings from PCEjin and VBjin, with one row per recorded frame. PC Engine recordings support up to five controller ports; buttons I and II have the IDs of A and B, and Run that of Start. VBjin recordings include both Virtual Boy direction pads: the left one has the IDs of the direction pad and the right one those of the N64's C-buttons.

Usage

```
read_mc2(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that

Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

`triggers` "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

`aliases` TRUE (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

`player()` returns the buttons for a selected port. The original command byte is retained in `commands`. Header fields identify the emulator and, when recorded, PC Engine CD mode. Incomplete binary packets and inconsistent controller fields produce an error.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_mc2(system.file("extdata", "legacy", "vboy.mc2", package = "replayr"))
# The Virtual Boy has two direction pads.
button_layout(movie)[, c("key", "name")]
```

---

read_mcm *Read a Mednafen rerecording movie*

Description

Imports MCM version 1 with fixed gamepad layouts for PC Engine, PC-FX, Atari Lynx, WonderSwan, Neo Geo Pocket, NES, and Master System. Each row represents one stored frame, with separate players for the format's controller ports. Numbered buttons have the lettered IDs in order: PC Engine and PC-FX I to VI are A, B, X, Y, L1 and R1, and Master System Fire 1 and 2 are A and B; `button_layout()` keeps their labels. Run, the Lynx's Pause and the Neo Geo Pocket's Option are Start; the Lynx's Option 1 and 2 are Select and Mode; PC Engine's switch between two and six buttons is Mode. The WonderSwan's X cross is the direction pad and its Y cross has the IDs of the N64's C-buttons. The Master System's Pause sits on the console and is returned by `system_channels()`.

Usage

```
read_mcm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

player() returns the digital input. commands retains the command byte, and source_packets retains complete packets. An incomplete final packet produces a warning and remains in source_tail. The ROM checksum bytes are preserved in header$romMD5Bytes. Peripheral layouts beyond these fixed gamepads require a separate decoder.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_mcm(system.file("extdata", "legacy", "pce.mcm", package = "replayr"))
n_players(movie)
# The numbered buttons of the PC Engine take the lettered identities in order.
button_layout(movie)[, c("key", "name")]
```

`read_mmv` *Read a Dega recording*

Description

Imports Master System and Game Gear input, including older MMV headers. Each row represents one complete stored frame. Master System recordings have two controller ports; Game Gear recordings have one player. Buttons 1 and 2 have the IDs of A and B and keep their labels in `button_layout()`. The Game Gear's Start is a button of the handheld. The Master System's Pause sits on the console and is returned by `system_channels()`.

Usage

```
read_mmv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's

controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

`triggers` "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

`aliases` TRUE (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

The header includes `declaredFrameCount`, `availableFrameCount`, `frameCountDifference`, `frameCount`, and region flags. Following Dega's playback rule, all complete packets through the end of the file are imported. `source_packets` retains packet bytes, including padding. An incomplete final packet produces a warning and remains in `source_tail`.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_mmv(system.file("extdata", "legacy", "sms.mmv", package = "replayr"))
# Buttons 1 and 2 are A and B; Pause is a button of the console.
button_layout(movie)[, c("key", "name")]
system_channels(movie)
```

`read_movie` *Import an emulator input recording*

Description

Dispatches by extension to the format-specific reader. Every reader returns compact digital columns, a stable cross-controller ID map, and original numeric channels. A row may be a frame, poll, input event, or tic depending on the format.

Usage

```
read_movie(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  format = NULL,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
format	Format of the file, as one of the extensions of supported_formats() ("fm2" or ".fm2"), or "atari_gc" or "atari_head" for a table of these data sets. The default NULL takes it from the extension of path, or from the content as described above. With format, a file called .zip is read as a recording of that format and not searched for recordings.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

The extension decides the reader. A file under another extension, or under none, is recognised by its content where its format has a signature: the first bytes of most binary formats, the `version 3` line of FCEUX, the members of a BizHawk or lsnes archive, the header of a BizHawk text movie or an openMSX replay, and the first line of a table of the Atari Grand Challenge data set or of Atari-HEAD (`read_txt()`). These two share the extension `.txt`, which says too little by itself: a `.txt` file is read as what its content shows. A file whose extension and signature disagree is refused, with the format its content shows. `format` names the format in every other case, for example for a Doom demo that is not called `.lmp`.

Value

A replayr_movie, or a replayr_poll_stream for unassigned controller ports or heterogeneous device polls, as in CTM. Use `button_layout()` for column identities, `player()` for logical states, and `stick_channels()` for selected stick fields.

Examples

```
path <- system.file("extdata", "sample.m64", package = "replayr")
movie <- read_movie(path, analog = "angle+magnitude")
button_layout(movie)
head(stick_channels(movie))
# A recording under another name: an SNES movie is known by its signature,
# a Doom demo has none and is named.
copy <- tempfile(fileext = ".dat")
file.copy(system.file("extdata", "sample.smv", package = "replayr"), copy)
read_movie(copy)
file.copy(system.file("extdata", "sample.lmp", package = "replayr"), copy, overwrite = TRUE)
read_movie(copy, format = "lmp")
unlink(copy)
```

`read_omr` *Read an openMSX replay*

Description

Reads gzip-compressed openMSX XML replay files. Supported event schemas cover keyboard matrices, joysticks, mice, trackballs, paddles, touchpads, light guns, and autofire settings. The initial keyboard state is read from the first replay snapshot. Every device is one controller: the keyboard first, then the devices on the joystick ports by name, as listed in `movie$header$controllers`. Their controls have the same IDs as in every other reader: a joystick is the direction pad with A and B, the buttons of a mouse or trackball are A and B, the JoyMega pad adds C, Start, X, Y, Z and Mode, a light-gun trigger and a paddle's button are A, and a touch on the touchpad is L3. `openmsx_key_r8_b0` identifies a keyboard matrix row and bit. All devices share one event stream, so each controller has a row for every event.

Usage

```
read_omr(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

Each row follows one replay event. movie$events$timestamp_ticks retains time in openMSX master-clock ticks (3,579,545 * 960 ticks per second), and timestamp_seconds expresses the same event times in seconds. timestamp_source, event_type, source_device, and source_xml retain the event's original time, type, device, and XML.

analog_channels() preserves the values of each device on its controller. Examples include mouse_dx and mouse_dy for movement and touch_x and touch_y for touch coordinates. A recorded autofire setting, such as openmsx_autofire_renshaturbo, belongs to the machine and is returned by system_channels(). Movement deltas apply to their event; coordinates and settings persist until updated. The replay schema version and machine name remain in movie$header$replay_version and movie$header$machine.

Value

A replayr_movie with event times, device controls, and source XML.

Examples

```
movie <- read_omr(system.file("extdata", "computers", "keyboard-joystick.omr", package = "replayr"))
# Every device is a controller.
movie$header$controllers
observation_time(movie)
```

---

read_pjm *Read a PJM recording*

Description

PSXjin text and binary recordings with standard or analogue controllers. Every complete record is imported, following PSXjin's file-length rule. The header retains declaredFrameCount, availableFrameCount, and their signed frameCountDifference. Text fields follow the declared encoding and controller types. Count differences remain metadata; incomplete records are rejected. Digital columns are compact; button_layout() gives their stable IDs. Cross, Circle, Triangle and Square keep their names and have the IDs of the SNES buttons in the same places: B, A, X and Y. Numeric source channels are always retained in analog_channels(). header$cdromIds lists the IDs of the discs played, in the order they were inserted, as PSXjin stores them (such as "SLUS00067").

Usage

```
read_pjm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.

analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_pjm(system.file("extdata", "sample.pjm", package = "replayr"))
movie
n_players(movie)
```

---

read_pxm *Read a PXM recording*

Description

PCSX standard digital and analogue-controller records. Unsupported devices are rejected; reserved standard-pad bits do not become stick clicks. Digital columns are compact; button_layout() gives their stable IDs. Cross, Circle, Triangle and Square keep their names and have the IDs of the

SNES buttons in the same places: B, A, X and Y. Numeric source channels are always retained in analog_channels(). header$cdromIds lists the IDs of the discs played, in the order they were inserted, as PCSX stores them (such as "SLUS00067").

Usage

```
read_pxm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_pxm(system.file("extdata", "semantic", "semantic-onehot.pxm", package = "replayr"))
# Cross, Circle, Triangle and Square take the identities of the buttons
# in the same places on an SNES pad.
button_layout(movie)[7:10, c("key", "name")]
channel_layout(movie)
```

`read_smv` *Read a SMV recording*

Description

Supported Snes9x versions and joypad controllers. Digital columns are compact; `button_layout()` gives their stable IDs. Numeric source channels are always retained in `analog_channels()`.

Usage

```
read_smv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that

Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

`triggers` "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

`aliases` TRUE (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

Each row is an input sample. A recording holds one sample more than its frame count. Version 4 and 5 files also declare a count of input samples; when that count is larger and the samples are stored, all of them are returned. Bytes after the declared range are left unread and counted in `header$unusedTrailingBytes`.

`header$romCRC32` and `header$romName` hold the CRC32 of the ROM and its internal name, as Snes9x stores them before the savestate; recordings of early Snes9x versions leave both empty. Japanese names keep their katakana.

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_smv(system.file("extdata", "sample.smv", package = "replayr"))
movie
press_counts(movie)
```

`read_tas` *Read a TAS recording*

Description

Early BizHawk text recordings.

Usage

```
read_tas(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

An imported recording containing original input fields and metadata.

Examples

```
movie <- read_tas(system.file("extdata", "expanded", "legacy-sms.tas", package = "replayr"))
movie
system_channels(movie)
```

`read_tasproj` *Read a TASPROJ recording*

Description

BizHawk TAS Studio projects.

Usage

```
read_tasproj(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Value

An imported recording containing original input fields and metadata.

Examples

```
path <- system.file("extdata", "expanded", "editor-input.tasproj",
  package = "replayr")
movie <- read_tasproj(path)
movie
as.data.frame(movie)
```

---

read_txt *Read Atari Grand Challenge and Atari-HEAD demonstration files*

Description

The Atari Grand Challenge and Atari-HEAD data sets hold human demonstrations of Atari 2600 games, collected for research on reinforcement learning. Both store the input as text tables with one row for each frame, in .txt files. The reader knows by the first line which of the two a file is:

Usage

```
read_txt(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.

deadzone — Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.

triggers — "raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.

aliases — TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

... — Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

- A trajectory of the Atari Grand Challenge data set, version 1 (Kurin et al. 2017, doi:10.48550/arXiv.1705.10998): one episode of play, recorded at 60 frames per second. The data set keeps the trajectories of each game in a folder of its own (spaceinvaders, qbert, mspacman, pinball, revenge); the file does not name its game.
- A trial of Atari-HEAD (Zhang et al. 2020, doi:10.1609/aaai.v34i04.6161; files at doi:10.5281/zenodo.3451402). The game waited for the player in every frame, and the table stores how long each frame was shown. The archive of each game holds the text files of its trials beside the archives of their screen images, and a trial can be read out of it: read_movie("breakout.zip", member = ...), with the members that movie_members() lists. The file meta_data.csv of the data set names the game of each trial.

Each row holds the action of its frame as one of the 18 action numbers of the Arcade Learning Environment. These are the states of one joystick, so player() returns its five controls: the four directions and the fire button, under the IDs an Atari 2600 joystick has in every other recording.

analog_channels() returns the action number as the table writes it (action) and, for a trial, how long each frame was shown (duration_ms, the column duration(ms)). A field that the table gives as null is NA; the last row of a trial has no action, so that no control is held in it. observation_time() of a trial is the time at which each frame appeared, the sum of the durations of the frames before it. movie$timing holds the frame numbers of a trajectory (frame) or the frame identifiers of a trial (frame_id), and movie$header$trajectoryId the number in the first line of a trajectory.

Only the input is read. The tables also give states of the game (the score, the reward of each frame, the episode and whether it ended) and, for a trial, the gaze positions of each frame: the points on

the screen at which the player was looking, measured by an eye tracker. These are not returned; `read.csv()` reads them from the same files.

`recording_info(movie)$format` says which table was read: `"atari_gc"` or `"atari_head"`. A text file that is neither is refused.

Value

A replayr_movie with one player.

See Also

`recording_format()` to tell the tables from other text files.

Examples

```
trajectory <- system.file("extdata", "tables", "atari-grand-challenge.txt", package = "replayr")
movie <- read_txt(trajectory)
colSums(player(movie))
head(analog_channels(movie))
trial <- read_txt(system.file("extdata", "tables", "atari-head.txt", package = "replayr"))
recording_info(trial)$format
observation_time(trial)
```

---

read_vbm *Read a VBM recording*

Description

VBA recordings for Game Boy, Game Boy Color, Super Game Boy and Game Boy Advance. Game Boy and Game Boy Color recordings have eight controls. Game Boy Advance and Super Game Boy recordings add L and R: a Super Game Boy is played with an SNES pad, and the format records both buttons. Digital columns are compact; `button_layout()` gives their stable IDs. Numeric source channels are always retained in `analog_channels()`.

Usage

```
read_vbm(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

Every stored controller record is imported; header$declaredFrameCount keeps the header's count. commands retains the bits of each controller word that lie outside the layout, with their original values: 1024 and 2048 mark a reset (old timing, and the timing used since format version 1.1), and 4096 to 32768 are the deprecated motion-sensor bits.

The header identifies the ROM by its internal title (header$romName) and its header checksum (header$romHeaderChecksum), and adds the global checksum of a Game Boy ROM (header$romGlobalChecksum) or the game code of a Game Boy Advance ROM (header$romGameCode, such as "A2CE"). In place of the global checksum, a Game Boy Advance recording holds the checksum of the BIOS, in header$biosChecksum.

Value

A replayr_movie with compact digital columns and selected stick fields.

Examples

```
movie <- read_vbm(system.file("extdata", "sample.vbm", package = "replayr"))
movie
recording_info(movie)
```

`read_wtf` *Read an Hourglass recording*

Description

Reads WTF files with the `66 54 77 02` signature and the equivalent HGM extension. Each frame stores up to eight Windows virtual-key codes; the first zero ends its list of held keys.

Usage

```
read_wtf(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

`path`	Recording file path.
`analog`	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
`deadzone`	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.

`triggers`	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
`aliases`	`TRUE` (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With `FALSE`, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
`...`	Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

Each row is a recorded frame. `movie$header$fps` contains the recorded nominal frame rate; `initial_time` and `clock_checks` retain the emulator's clock metadata. The header also includes the keyboard layout, executable name, and Hourglass revision.

`button_layout()` identifies keys by their hexadecimal virtual-key code: for example, `windows_vk_25` is Left and `windows_vk_41` is A. `movie$source_keycodes` retains all eight source bytes for every frame, and `movie$source_header` retains the original 1024-byte header.

Value

A replayr_movie with keyboard states and source metadata.

Examples

```
movie <- read_wtf(system.file("extdata", "computers", "keyboard-jump.wtf", package = "replayr"))
# Keyboard keys keep the codes of the recording.
button_layout(movie)[, c("id", "name")]
as.data.frame(movie)
```

`read_ymv` *Read a Yabause rerecording movie*

Description

Imports enhanced YMV version-1 text and binary recordings for one Saturn gamepad. Each row represents a recorded frame. The thirteen digital fields include four directions, Start, six face buttons, and two shoulder buttons.

Usage

```
read_ymv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: "raw" (default), "cardinal4", "cardinal8", "angle", or "angle+magnitude". A scalar applies to both sticks; c(left = "cardinal8", right = "angle") selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through stick_channels(). Original X/Y values remain in analog_channels() in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. NULL uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (PAD_Clamp in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an NA angle. Magnitude is kept.
triggers	"raw" (default) or "digital". For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	TRUE (default) places the controls of a recording under the shared IDs that button_layout() describes. With FALSE, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.
...	Reader-specific arguments, such as controller_order for M64 or DTM, or member for a recording inside a ZIP archive (see movie_members()).

Details

player() returns button states, button_layout() identifies the controls, and commands retains the source command byte. Earlier YMV layouts that precede this version-1 format are unsupported.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_ymv(system.file("extdata", "legacy", "saturn.ymv", package = "replayr"))
movie
recording_info(movie)
```

read_zmv *Read an enhanced ZSNES recording*

Description

Imports enhanced ZMV recordings for up to five SNES gamepads. Repeat records are expanded into frame rows. `player()` returns a selected pad, and `button_layout()` identifies its twelve buttons.

Usage

```
read_zmv(
  path,
  analog = "raw",
  deadzone = NULL,
  triggers = "raw",
  aliases = TRUE,
  ...
)
```

Arguments

path	Recording file path.
analog	Stick representation: `"raw"` (default), `"cardinal4"`, `"cardinal8"`, `"angle"`, or `"angle+magnitude"`. A scalar applies to both sticks; `c(left = "cardinal8", right = "angle")` selects them independently. Cardinal modes add separate left/right logical direction columns. Continuous modes return numeric fields through `stick_channels()`. Original X/Y values remain in `analog_channels()` in every mode. Absent sticks add no columns.
deadzone	Radial threshold in centred axis half-range units, from zero to one. One value applies to both sticks; named left/right values are allowed. `NULL` uses the default of the controller, whichever emulator recorded it: 0.125 for the sticks of Nintendo's consoles (Nintendo 64, GameCube, Wii extensions, 3DS) and 0.25 for all other sticks. The first is close to the 15 of 128 units that the GameCube's controller library discards on each axis (`PAD_Clamp` in libogc). The second covers the resting position of a released PlayStation stick, up to about 32 of 128 units from the centre (psx-spx), and is close to the 7849 and 8689 of 32767 that Microsoft gives for the thumbsticks of PC gamepads (XInput). A game applies a threshold of its own, which the recording does not store; give that value to study one game. Neutral observations have no direction and an `NA` angle. Magnitude is kept.
triggers	`"raw"` (default) or `"digital"`. For DTM, digital additionally projects each analogue trigger at 128/255 into separate L2/R2 columns. Hardware L/R click bits remain distinct from thresholded analogue triggers.
aliases	`TRUE` (default) places the controls of a recording under the shared IDs that `button_layout()` describes. With `FALSE`, a control that would be placed under another control's ID keeps the key of its source, so every control of the recording has a column of its own and no two are combined.

`...` Reader-specific arguments, such as `controller_order` for M64 or DTM, or `member` for a recording inside a ZIP archive (see `movie_members()`).

Details

Reset commands are attached to the following frame. Chapter offsets and counts remain in `chapter_offsets` and the header. A header frame-count mismatch produces a warning, with complete stored frames retained. The header also preserves the PAL flag and any reset after the last frame. `header$romCRC32` is the CRC32 that ZSNES computed for the ROM. Mouse and Super Scope layouts require separate peripheral decoding.

Value

A replayr_movie with compact digital columns and source metadata.

Examples

```
movie <- read_zmv(system.file("extdata", "legacy", "holds.zmv", package = "replayr"))
movie
as.data.frame(movie)
```

---

recording_format *Format of a recording file*

Description

Says as which format `read_movie()` reads a file, without reading it: by its extension, or by its content for a file under another extension and for the extension `.txt`, which two formats share and many files that are no recordings carry. Use it to pick the recordings out of a folder.

Usage

```
recording_format(path)
```

Arguments

`path` File paths.

Value

Character vector with one entry per path: the format as the argument `format` of `read_movie()` takes it (an extension without its dot, or `"atari_gc"` or `"atari_head"`), and `NA` for a file that `read_movie()` does not read as a recording, for a folder and for a path that does not exist. A ZIP archive that holds recordings is no recording itself and gives `NA`; see `movie_members()`.

See Also

`supported_formats()`

Examples

```
files <- list.files(system.file("extdata", package = "replayr"), full.names = TRUE)
data.frame(file = basename(files), format = recording_format(files))
# Of the text files, the tables of the two Atari data sets are recordings.
texts <- list.files(system.file("extdata", package = "replayr"), "[.]txt$",
  recursive = TRUE, full.names = TRUE)
data.frame(file = basename(texts), format = recording_format(texts))
```

---

recording_info *Platform, timing and rerecord count of a recording*

Description

Formats name these properties differently and keep them in different places (`Platform`, `system`, `palFlag`, `isPal`, `fps`, `frameRate`, `rerecordCount`, `rerecords`); `movie$header` holds them under the names of the format. This function reports them under one name each.

Usage

```
recording_info(movie, fallback_rate = NA)
```

Arguments

`movie`	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.
`fallback_rate`	Frame rate, in frames per second, to use where neither the recording nor its platform gives one, such as 60 or 50 for a Nintendo 64 recording. The default `NA` assumes none.

Details

`platform` is BizHawk's code for the system (`NES`, `SNES`, `GB`, `GBC`, `GBA`, `N64`, `PSX`, `SAT`, `GEN`, `SMS`, `GG`, `PCE`, `NDS`, `3DS`, `A26` and so on), also for recordings of other emulators, with `GC` and `Wii` for Dolphin and `PC` for recordings of computer programs. A format that serves one console implies its platform.

`frame_rate` is in video frames per second, and `frame_rate_source` says where it comes from: `"recording"` for a rate the file states, `"format"` for the 35 tics per second of Doom engines and the 60 frames per second of the Atari Grand Challenge data set, and `"platform"` for the nominal rate of the console and video standard, as BizHawk uses it to time recordings. The nominal rate is that of the original hardware. No rate is assumed where the platform or, for a console sold in both standards, the video standard is not known, unless `fallback_rate` names one; the source is then `"fallback"`.

Value

A data frame with one row: `format`, `platform`, `video_standard` (`"NTSC"`, `"PAL"` or `NA`), `unit` (see `observation_unit()`), `observations`, `frame_rate`, `frame_rate_source`, `duration` in seconds (from the first observation to the end of the last; `NA` where `observation_time()` is) and `rerecords`.

See Also

observation_time()

Examples

```
movie <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
recording_info(movie)
# A Nintendo 64 recording in BizHawk's older format names no rate.
n64 <- read_movie(system.file("extdata", "expanded", "legacy-n64.bkm", package = "replayr"))
recording_info(n64)[c("frame_rate", "frame_rate_source", "duration")]
recording_info(n64, fallback_rate = 60)[c("frame_rate", "frame_rate_source", "duration")]
```

---

replayr_movie *The* replayr_movie *S3 class*

Description

An imported recording with assigned input streams. Recordings whose polls have unassigned ports use replayr_poll_stream. A replayr_movie is a list with the following elements:

Details

header Named list of header key-value pairs lifted from the source file. The synthetic _format entry records which parser produced the movie, such as "fm2", "bk2", or "ltm".

commands Integer vector with one entry per observation. Per-observation emulator command bits, where the format uses them.

inputs Logical 3D array, dimensions (observations, players, buttons). TRUE means the button was held at that observation.

analog List of numeric channel matrices, one per player. Rows align with inputs; columns are named by the keys of channel_layout(), and units depend on the format.

analog_mode Selected representation label. Numeric source values are retained in every mode.

stick_modes Named left/right choices: "raw", "cardinal4", "cardinal8", "angle", or "angle+magnitude".

button_ids, button_keys, layout_version Stable control identities and registry version for the compact input columns.

stick_values Selected numeric stick channels, one matrix per player; accessed with stick_channels().

source_path Path the movie was loaded from, or NA. ZIP imports also retain the selected source_member.

events, timing, input_events Where present, source-specific event records and timing. See the format-specific reader.

button_mnemonics Character vector. One mnemonic per third-dimension slot of inputs; source-defined controls may use multi-character labels.

button_names Character vector. Friendly labels paired with button_mnemonics.

The accessors name control columns by button_keys; mnemonics and labels are further columns of button_layout().

Use n_observations(), n_players(), n_buttons() to read the array shape and player() to extract one player's buttons as a 2D matrix. summary() counts held observations and activations per control, as.data.frame() returns one row per observation, and [movie[i]][subset-recording] selects observations and keeps the class.

See Also

read_movie(), universal_buttons()

Examples

```
movie <- read_movie(system.file("extdata", "sample.dtm", package = "replayr"))
class(movie)
# Observations, controllers and controls.
dim(movie$inputs)
names(movie$header)[1:5]
movie
# The accessors and methods work on a selection as on the whole recording.
summary(movie[1:600])
```

---

replayr_poll_stream *Ordered controller and device polls*

Description

Multi-controller M64 and DTM files store ordered samples without per-sample port IDs. The S3 class replayr_poll_stream retains the decoded controls and original bytes. polls() returns one row per poll, with port = NA_integer_, digital controls, source numeric channels, selected stick fields, and command bits. DTM and CTM streams also expose format-specific poll flags. button_layout() maps digital keys to stable IDs. analog_channels() and stick_channels() select the single stored stream with idx = 1.

Details

n_observations() counts polls; n_players() returns NA_integer_. player() requires an assigned controller and therefore directs callers to polls(). If emulator evidence establishes a repeating polling order, supply controller_order to read_m64() or read_dtm() to construct a replayr_movie. The format-specific help explains how player indices map to physical controller ports.

CTM stores separate pad, touch, and auxiliary-device events. Its observation flags identify which controls were sampled; observed_inputs provides the per-control mask. Mixed DTM adds device family and physical port when known. Wii reports marked valid_report = FALSE preserve missing source payloads.

See Also

read_m64, read_dtm

Examples

```
# Two GameCube controllers are enabled, and the file does not say which
# poll belongs to which.
stream <- read_movie(system.file("extdata", "paper", "unassigned.dtm", package = "replayr"))
class(stream)
stream
n_players(stream)
button_layout(stream)$key
polls(stream)[, c("poll", "port", "a", "left_x", "left_y")]
```

---

stick_channels *Selected stick directions, angles and magnitudes*

Description

Angles are degrees anticlockwise from right: right = 0, up = 90, left = 180, down = 270. Neutral angles and missing observations are NA. Cardinal columns contain the nearest direction's centre angle. Magnitude is Euclidean length after dividing each centred source axis by its declared half-range; diagonal magnitudes can exceed one. It is not a calibrated force measurement. `angle` excludes magnitude; `angle+magnitude` includes it. Raw source X/Y values remain in `analog_channels()` for every mode. The Saturn BK2 3D-pad fields `Stick Horizontal` and `Stick Vertical` use unsigned byte axes centred at 128, with smaller vertical values up. Analogue shoulder values remain separate source channels.

Usage

```
stick_channels(movie, idx = 1L)
```

Arguments

`movie`	A replayr_movie or replayr_poll_stream returned by `read_movie()` or a format-specific reader.
`idx`	Player index, 1-based. Defaults to 1 (player 1).

Value

Numeric matrix with selected fields. Absent sticks have no columns.

Examples

```
path <- system.file("extdata", "sample.m64", package = "replayr")
movie <- read_movie(path, analog = "angle+magnitude")
head(stick_channels(movie))
```

`subset-recording` *Select observations of a recording*

Description

`movie[i]` keeps the observations `i` and returns a recording of the same class, so that every accessor works on the selection as on the whole recording. Whatever is stored once per observation is cut together: button states, numeric channels, commands, console controls, poll fields, and the tables of a reader that follow the observations (timing, events, subframes, source bytes).

Usage

```
## S3 method for class 'replayr_movie'
x[i, ...]

## S3 method for class 'replayr_poll_stream'
x[i, ...]
```

Arguments

`x`	A replayr_movie or replayr_poll_stream.
`i`	Observations to keep: positions, negative positions to leave out, or a logical vector with one entry per observation.
`...`	Not used. A recording is selected by observation only; use `player()` for the buttons of one controller.

Details

The header and the tables that describe the file as a whole stay as they are, so header counts still refer to the file. `observation_index` in the result gives the position in the file of each observation kept.

Value

A recording of the class of `x`.

Examples

```
movie <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
first_ten_seconds <- movie[1:600]
n_observations(first_ten_seconds)
# The observations with A held.
n_observations(movie[player(movie)[, "a"]])
```

summary.replayr_movie *Summary of a recording*

Description

Describes each control and numeric channel of a recording. For a control, `held` is the number of observations in which it is active and `activations` the number of times it becomes active, counting a control that is active in the first observation. A button held for a long time has many observations and one activation; a button tapped repeatedly has many activations.

Usage

```
## S3 method for class 'replayr_movie'
summary(object, ...)

## S3 method for class 'replayr_poll_stream'
summary(object, ...)
```

Arguments

`object`	A replayr_movie or replayr_poll_stream.
`...`	Not used.

Value

An object of class `summary.replayr_movie`: a list with the `format`, the `unit` of observation, the number of `observations` and `controllers`, and two data frames. `controls` has one row per controller and control (`controller`, `key`, `name`, `held`, `activations`); `channels` has one row per controller and numeric channel (`controller`, `key`, `name`, `min`, `max`, `missing`). Printing shows the controls that are active at least once.

Examples

```
movie <- read_movie(system.file("extdata", "sample.fm2", package = "replayr"))
summary(movie)
summary(movie)$controls
```

supported_formats *Supported recording formats*

Description

The package reads 37 file extensions in 33 formats. Five extensions have the file structure of another format. One extension, `.txt`, is that of two formats, those of the Atari Grand Challenge and Atari-HEAD data sets, which `read_movie()` tells apart by their first line. Any other `.txt` file is none of them; `recording_format()` says which files of a folder are recordings.

Usage

```
supported_formats(details = FALSE)
```

Arguments

details	FALSE (default) returns the file extensions. TRUE returns a table of the formats.

Value

Character vector of file extensions in lower case with a leading dot. With details = TRUE, a data frame with one row per extension and format (two rows for .txt): extension, structure (the extension that names the file structure: the extension itself, except that TAStudio projects and GBA-Hawk recordings are BK2 archives, early BizHawk TAS recordings are BKM files, FCEUX TAS Editor projects begin with an FM2 recording, and HGM is a second extension of Hourglass WTF; the two formats under .txt have the names atari_gc and atari_head), reader (the function that reads it; read_movie() calls it for a file of that format), emulator (the emulator or program that writes such files, with its game in brackets where it serves a single game, or the data set whose tables they are) and platform (the platform code of recording_info() where the format serves one console, otherwise NA: the recording names its platform).

Examples

```
supported_formats()
head(supported_formats(details = TRUE))
```

---

system_channels *Controls of the console or cabinet*

Description

Returns the controls that belong to the machine and not to a player's controller: reset and power, coin slots and service buttons, disc and tape controls, the lid of a handheld, and numeric settings such as a selected disc. They never take a column in a player's button layout.

Usage

```
system_channels(movie)
```

Arguments

movie	A replayr_movie.

Details

Columns are named by key, as in player() and analog_channels(): the name of the source in lower case with underscores, so that reset, power, pause and coin_1 are the same column in every format that records them. system_layout() gives the label each column has in the recording.

Value

Data frame with one row per observation; zero columns for formats without separately decoded controls of the system.

Examples

```
path <- system.file("extdata", "sample.bk2", package = "replayr")
head(system_channels(read_movie(path)))
```

`system_layout` *Names of the controls of the console or cabinet*

Description

Names of the controls of the console or cabinet

Usage

```
system_layout(movie)
```

Arguments

`movie`	A replayr_movie.

Value

Data frame with one row for each column of `system_channels()`: `column`, `key` (the column name) and `name` (the label of the recording or of its format).

Examples

```
path <- system.file("extdata", "sample.bk2", package = "replayr")
system_layout(read_movie(path))
```

`universal_buttons` *Universal controller mnemonics*

Description

Returns the mnemonic column of `button_layout()`: one character for each registered control, for writing a controller state as compact text. Controls outside the registry have mnemonics such as `{key_space}`. The accessors name their columns by key, not by mnemonic; use `button_layout()` on a movie to obtain its complete layout.

Usage

```
universal_buttons()
```

Value

Character vector.

Examples

```
head(universal_buttons(), 12)
# A controller state as text: Right, B and A held.
layout <- button_layout()
paste(layout$mnemonic[match(c("dpad_right", "b", "a"), layout$key)], collapse = "")
```

`universal_button_names` *Universal controller names*

Description

Universal controller names

Usage

```
universal_button_names()
```

Value

Character vector in stable registry order.

See Also

`button_layout()`

Examples

```
head(universal_button_names(), 12)
# The registry, with the ID and key of each name.
head(button_layout()[, c("id", "key", "name")], 12)
```

Index

[.replayr_movie *(*subset-recording*)*, 81
[.replayr_poll_stream *(*subset-recording*)*, 81

align_buttons, 3
align_buttons(), *13*
analog_channels, 4
analog_channels(), *8*, *15*, *16*, *18*, *19*, *21–24*, *26–29*, *32–50*, *52*, *53*, *55*, *56*, *58*, *60–64*, *66–72*, *74*, *75*, *79*, *80*, *83*
as.data.frame(), *79*
as.data.frame.replayr_movie, 5
as.data.frame.replayr_poll_stream *(*as.data.frame.replayr_movie*)*, 5

base::as.data.frame(), *5*
button_layout, 6
button_layout(), *3*, *5*, *8*, *9*, *13–16*, *18*, *20*, *21*, *23*, *24*, *26–30*, *32–42*, *44*, *45*, *47–52*, *54–66*, *68–71*, *73–75*, *79*, *84*, *85*

channel_layout, 8
channel_layout(), *4*, *5*, *14*, *78*

movie_members, 9
movie_members(), *16*, *20*, *21*, *23*, *24*, *26*, *28*, *30*, *32*, *34*, *35*, *37–39*, *41*, *42*, *44*, *45*, *47*, *49*, *51*, *52*, *54*, *55*, *57*, *58*, *60*, *62*, *63*, *65*, *66*, *68*, *69*, *71*, *73*, *74*, *76*

n_buttons, 9
n_buttons(), *79*
n_observations, 10
n_observations(), *12*, *47*, *79*
n_players, 11
n_players(), *79*

observation_time, 11
observation_time(), *24*, *69*, *77*, *78*
observation_unit, 12
observation_unit(), *10*, *12*, *77*

player, 13
player(), *4*, *8*, *15*, *24*, *27*, *32*, *37*, *41*, *45*, *54*, *55*, *59*, *69*, *74*, *75*, *79*, *81*, *83*
polls, 14
polls(), *5*, *13*, *22*, *79*
press_counts, 14
press_counts(), *13*

read_bk2, 15
read_bk2(), *19*
read_bk2_chunks, 17
read_bk2_chunks(), *15*
read_bkm, 19
read_ctas, 20
read_ctm, 22
read_dem, 23
read_dft, 25
read_dsm, 27
read_dtm, 28
read_dtm(), *14*, *79*
read_fbm, 31
read_fcm, 33
read_fm2, 34
read_fm3, 36
read_fmv, 37
read_gbmv, 38
read_gmv, 40
read_gzm, 41
read_jrsr, 43
read_lmp, 44
read_lsmv, 46
read_ltm, 48
read_m64, 50
read_m64(), *79*
read_mar, 51
read_mc2, 53
read_mcm, 54

read_mmv, 56
read_movie, 57
read_movie(), *4*, *8–13*, *15*, *76, 77*, *79, 80*, *82, 83*
read_omr, 59
read_pjm, 61
read_pxm, 62
read_smv, 64
read_tas, 65
read_tasproj, 67
read_txt, 68
read_txt(), *59*
read_vbm, 70
read_wtf, 72
read_ymv, 73
read_zmv, 75
recording_format, 76
recording_format(), *70*, *82*
recording_info, 77
recording_info(), *11, 12*, *83*
replayr_movie, *3–5*, *8–13*, *15*, *17*, *21*, *25, 26*, *28*, *31, 32*, *34, 35*, *38*, *41, 42*, *44*, *46*, *48–51*, *54*, *56, 57*, *59*, *61, 62*, *64, 65*, *70, 71*, *73, 74*, *76, 77*, 78, *79–84*
replayr_poll_stream, *4, 5*, *8–15*, *23*, *28*, *31*, *50, 51*, *59*, *77, 78*, 79, *80–82*

stick_channels, 80
stick_channels(), *16*, *18, 19*, *21, 22*, *24*, *26, 27*, *29*, *32, 33*, *35–37*, *39, 40*, *42, 43*, *45*, *47*, *49, 50*, *52, 53*, *55, 56*, *58–60*, *62–64*, *66–68*, *71, 72*, *74, 75*, *78, 79*
subset-recording, 81
summary(), *15*, *79*
summary.replayr_movie, 82
summary.replayr_poll_stream *(*summary.replayr_movie*)*, 82
supported_formats, 82
supported_formats(), *58*, *76*
system_channels, 83
system_channels(), *7*, *32*, *46*, *54*, *56*, *61*, *84*
system_layout, 84
system_layout(), *83*

universal_button_names, 85
universal_buttons, 84
universal_buttons(), *79*