LUDOpy

Release 1.1.2

Apr 05, 2020

Contents:

CHAPTER 1

LUDOpy

1.1 ludopy package

1.1.1 Submodules

1.1.2 ludopy.game module

class `ludopy.game.`**`Game`**

Bases: `object`

The Game. This class is the only needed class for normal use

`all_players_finish``()`

Returns rather all players has finish

Return allFinish Bool rather all players has finish the game

Rtype allFinish bool

`answer_observation`(*piece_to_move*)

Answers a observation. A observation has to be given before a answer can be given.

Parameters **`piece_to_move`** (*`int`*) – Which piece to move. If there was no pieces there cloud be moved the parameter is ignored

Return obs Who the game was after the given move was done. obs is: (dice, move_pieces, player_pieces, enemy_pieces, player_is_a_winner, there_is_a_winner)

Rtype obs (int, list with upto 4 int's, list with 4 int's, list of 4 lists with 4 int's, bool, bool)

`get_hist``()`

Returns the history there has been recorded during the game. This history can be used to make a video of the game. The history will have been extended when a observation was given and when a answer to a observation was given.

Return hist list of [pieces, current_dice, first_winner_was, current_player, round]

Rtype hist [list with 4 lists with 4 int's, int, bool, int, int]

get_observation()

Return the state/observation of the game and which players turn it is A given observation has to be answered before a new one can be given.

Returns

- obs: The observation taken of the state of the game seen from the player given in the return current_player (dice, move_pieces, player_pieces, enemy_pieces, player_is_a_winner, there_is_a_winner)
- current_player: Which players turn it is

Rtype obs (int, list with upto 4 int's, list with 4 int's, list of 4 lists with 4 int's, bool, bool)

Rtype current_player int

get_piece_hist(*mode=0*)

Will return the how the pieces was recorded during the game.

Parameters **mode** (*int*) – 0: All recorded pieces is returnt. 1: Only if a change is done there will be a new set of pieces. 2: Only unique set of pieces (order is preserved)

Return piece_hist List of sets of pieces [player 1, player 2, player 3, player 4]

Rtype piece_hist list of 4 lists with 4 int's

get_pieces(*seen_from=None*)

Returns the pieces places on the board

Parameters **seen_from** (*int*) – indicate which player the pieces and enemy pieces are seen from. If None then the pieces from all 4 player are given and no enemy pieces

Returns

- pieces: The pieces for alle the players (if seen_from = None) else the pieces for the player given in seen_from
- enemy_pieces: The pieces of the enemys if a player is given in seen_from

Rtype pieces list of 4 int's

Rtype enemy_pieces list with 4 lists each with 4 int's

get_winner_of_game()

Returns the winner of the game

Return winner If there has been a winner the winner is return if not -1 is returned

Rtype winner bool

get_winners_of_game()

Returns the winners of the game

Return gameWinners A list of the winners of the game in the order they got all piece in goal

Rtype gameWinners list with upto 4 int's

render_environment()

Will render the last record in the history

Return board_img A image of the board

Rtype board_img ndarray, RGB colorspace

reset()
Resets the game and the game history

save_hist(*file_name*)
Saves the history of the game as a npy file

Parameters **file_name** (*str*) – The file name to save under. Has to have the .npy (numpy file) extension

save_hist_video(*video_out*, *fps=8*, *frame_size=None*, *fourcc=None*)
Saves a video of the game history

Parameters

- **video_out** (*str*) – The file name to save under
- **fps** (*float*) – Frames per second
- **frame_size** (*tuple*) – The frame size to save in (width, height). If None is given the full board size is used
- **fourcc** (*str*) – FourCC code to be used. If None is given the FourCC code will be tried to create fro the file extension (works on .mp4 and .avi)

1.1.3 ludopy.player module

class ludopy.player.**Player**
Bases: object

A class used by the Game class. This class is not needed for normal use

get_pieces()
Returns the players pieces

Return pieces The players pieces

Rtype pieces list

get_pieces_that_can_move(*dice*)
Return the pieces that can move with the given dice

Parameters **dice** (*int*) – The dice the move will be done with

Returns movable_pieces: A list with the pieces that can be moved

Rtype movable_pieces list

move_piece(*piece*, *dice*, *enemys*)
Move the players piece the given dice following the game rules. Returns the new locations of the enemy's pieces

Parameters

- **piece** (*int*) – The piece to move
- **dice** (*int*) – The dice to make the move with
- **enemys** (*list with 4 lists each with 4 int's*) – The enemy's pieces

Return enemys The new locations of the enemy's pieces

Rtype enemys list with 4 lists each with 4 int's

player_winner()
Returns rather the player is a winner or not

Returns winner: A bool that indicate rather the player is a winner or not

Rtype winner bool

set_all_pieces_to_home()
Sets all the players pieces to the home index

set_pieces(*pieces*)
Sets the players pieces

Parameters **pieces** – The pieces to set the players pieces to

ludopy.player.**enemy_pos_at_pos**(*pos*)
Returns the index's the other players has to be in to be in the same location as the one given in pos

Parameters **pos** (*int*) – The location to check for

Return enemy_pos The locations the enemy's pieces has to be at

Rtype enemy_pos list with 4 int's

ludopy.player.**get_enemy_at_pos**(*pos*, *enemys*)
Returns the enemy's and the pieces they have at the given location

Parameters

- **pos** (*int*) – The location to check for
- **enemys** – The locations for the enemy's pieces in a list of 4 lists

Returns

- enemy_at_pos: The enemy's there are at the location
- enemy_pieces_at_pos: The pieces the enemy's has at the location

Rtype enemy_at_pos list

Rtype enemy_pieces_at_pos list of list

1.1.4 ludopy.visualizer module

ludopy.visualizer.**draw_basic_board**(*draw_taile_number=False*)

ludopy.visualizer.**draw_dice**(*board*, *dice*, *player*)

ludopy.visualizer.**draw_dice_backgound**(*board*)

ludopy.visualizer.**draw_moment**(*board*, *moment*)

ludopy.visualizer.**draw_move_count**(*board*, *count*)

ludopy.visualizer.**draw_move_count_backgound**(*board*)

ludopy.visualizer.**draw_multi_box**(*board*, *top_left_taile*, *bottom_right_taile*, *line_color=None*, *fill_color=None*, *thickness=2*)

ludopy.visualizer.**draw_piece**(*board*, *n*, *m*, *amount*, *color*, *thickness=5*, *lineType=8*, *shift=0*)

ludopy.visualizer.**draw_players**(*board*, *player_pieces*)

ludopy.visualizer.**draw_tail**(*img*, *n*, *m*, *line_color=None*, *fill_color=None*, *thickness=2*)

`ludopy.visualizer.`**`draw_taile_indxs`**(*board*)

`ludopy.visualizer.`**`draw_text`**(*board*, *text*, *center*, *color*, *thickness=1*, *fontScale=0.5*)

`ludopy.visualizer.`**`get_all_tailes_within`**(*n_start*, *n_end*, *m_start*, *m_end*)

`ludopy.visualizer.`**`get_taile_cord`**(*n*, *m*)

`ludopy.visualizer.`**`get_tailes`**(*player_pieces*)

`ludopy.visualizer.`**`get_tailes_player`**(*player_pieces*, *player*)

`ludopy.visualizer.`**`make_img_of_board`**(*pieces*, *dice*, *players_dice*, *round_number*)

`ludopy.visualizer.`**`make_video_from_hist_file`**(*hist_file*, *video_out*, *fps=8*, *frame_size=None*, *fourcc=None*)

`ludopy.visualizer.`**`put_image_at_taile`**(*board*, *image*, *n*, *m*, *mask=None*)

`ludopy.visualizer.`**`save_hist_video`**(*filename*, *hist*, *fps=8*, *frame_size=None*, *fourcc=None*)

1.1.5 Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

CHAPTER 3

GitHub

GitHub at: https://github.com/SimonLBSoerensen/LUDOpy

Python Module Index

l

Index