Game_app module

class Game_app.**Game**(*w*, *h*, *title='None'*) [source]

Bases: **Gui.Gui_Window**

Class for the game.

> **Note:**
>
> This function contains all the game gui logic

Draw_accepting_page() [source]

Gui logic for accepting challenge page

Draw_challange_page() [source]

Gui logic for challenge page

Draw_game() [source]

Gui logic for the game stage, it shows the grid of the game and let the player draw the symbol in it with the fingers untill the game is over. Its only agains the AI player.

Draw_game_online() [source]

Gui logic for the game stage, it shows the grid of the game and let the player draw the symbol in it with the fingers untill the game is over. Its versus a real player and it works with networking.

Draw_menu() [source]

Gui logic for the main menu of the game, it shows the list of players the button to start the game and the cursor of the finger.

Draw_test() [source]

context() [source]

It process the conext of the game and show the specific game stage.

drawGrid() [source]

Draw the board grid 3x3 in the gui window.

draw_x(*y*, *r*) [source]

Draw an X at the gui window in the coordinate. :param x: x coordinate :type name: int :param y: y coordinate :type name: int :param r: r length :type name: int

handle_inbox() [source]

It handle the incoming messages when the multiplayer game is on

set_frame(*frame*) [source]

Set the actual frame of the GUI.

Parameters: **frame** – frame to set

start_loop() [source]

Main loop of the game GUI, it calls every time to render_frame untill the game finishes

Game_app.**hand_button**(*label*, *x*, *y*, *sx*, *sy*, *cursore=(0, 0)*) [source]

It displays a rectangle button that can change the color when the cursor is inside :returns: bool – If the cursor is inside of the rectangle

Game_app.**is_over**(*min*, *max*, *pos*) [source]

Check if the pos coordinates is inside min and max coordinates :returns: bool – if the pos is inside the rectangle