



FACE Recognition System ...

Monika Grądzka
Robert Kazimierczak

MENU PROGRAMU

REGISTRATION
System

HELP - C

Type:

BUTTON Description

UP Move the camera up

DOWN Move the camera down

LEFT Move the camera left

RIGHT Move the camera right

H Move the camera to home position

P Take pictures

C Show command list

ESC Exit the application

Type:

Type directory name where the pictures will be stored:

Super Mario

Type number of pictures that will be made:

5

wait...

Error! No screenshots were taken.

Type:

Type directory name where the pictures will be stored:

Super Mario

Type number of pictures that will be made:

5

wait...

Success! Your screenshot/s in dir Super Mario.

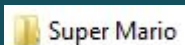
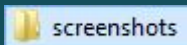
Type:


UŻYTE BIBLIOTEKI


- **libcurl** - komunikacja z kamerą poprzez komunikaty HTTP,
- **openCV** - zapis klatek do plików PNG


ZDJĘCIA NA ŻĄDANIE


- /screenshots/imię nazwisko
- nazwa zdjęcia: "%F %H-%M-%S"
- liczba zdjęć jako parametr




 2017-04-05 01-10-27

 2017-04-05 01-10-29

 2017-04-05 01-10-32

 2017-04-05 01-10-28

 2017-04-05 01-10-30



OBSŁUGA KLAWIATURY

```
void t_keyboard(Camera *camera)
{
    bool state = 0;
    while (!state)
    {
        checkKeyboard(camera, VK_UP);           //up
        checkKeyboard(camera, VK_DOWN);         //down
        checkKeyboard(camera, VK_LEFT);         //left
        checkKeyboard(camera, VK_RIGHT);        //right
        checkKeyboard(camera, VK_HOME);         //home position (H)
        checkKeyboard(camera, VK_PICTURE);      //take picture (P)
        checkKeyboard(camera, VK_COMMAND);      //command list
        state=checkKeyboard(camera, VK_ESCAPE); //exit
    }
}
```

```
bool checkKeyboard(Camera *camera, int key)
{
    if (GetAsyncKeyState(key) & 0x8000) //change sign of number
    {
        if (key == VK_ESCAPE) { return 1; }
        if (key == VK_COMMAND) { Utilities::printCommandList(); }
        if (key == VK_PICTURE) { Utilities::cleanBuffer(); }
        camera->sendMessage(key);

        while (GetAsyncKeyState(key) & 0x8000) { }
        if (key < VK_LEFT || key > VK_DOWN) { return 0; }
        camera->sendMessage(0);
    }
    return 0;
}
```

KOMUNIKACJA Z KAMERĄ

rozpocznij ruch w **dół**

22955	144.460659	192.168.1.113	192.168.1.116	HTTP	432 GET /hy-cgi/ptz.cgi?cmd=ptzctrl&act=down HTTP/1.1
-------	------------	---------------	---------------	------	---

rozpocznij ruch w **górę**

22407	142.921556	192.168.1.113	192.168.1.116	HTTP	428 GET /hy-cgi/ptz.cgi?cmd=ptzctrl&act=up HTTP/1.1
-------	------------	---------------	---------------	------	---

rozpocznij ruch w **lewo**

9104	66.702208	192.168.1.113	192.168.1.116	HTTP	432 GET /hy-cgi/ptz.cgi?cmd=ptzctrl&act=left HTTP/1.1
------	-----------	---------------	---------------	------	---

rozpocznij ruch w **prawo**

3927	39.342483	192.168.1.113	192.168.1.116	HTTP	434 GET /hy-cgi/ptz.cgi?cmd=ptzctrl&act=right HTTP/1.1
------	-----------	---------------	---------------	------	--

zatrzymaj ruch kamery

22481	143.369799	192.168.1.113	192.168.1.116	HTTP	432 GET /hy-cgi/ptz.cgi?cmd=ptzctrl&act=stop HTTP/1.1
-------	------------	---------------	---------------	------	---

powrót do pozycji **domowej**

34035	185.236062	192.168.1.113	192.168.1.116	HTTP	432 GET /hy-cgi/ptz.cgi?cmd=ptzctrl&act=home HTTP/1.1
-------	------------	---------------	---------------	------	---