

Dokumentacja Projektu –

Gra w „Kółko i krzyżyk” za pomocą kamery
(przechwytywanie obrazu)

Przedmiot: Podstawy Teleinformatyki
Prowadzący: mgr inż. Przemysław Walkowiak

Wykonał:
Piotr Marciniak (nr indeksu: 112656)
Informatyka – Rok III

30 czerwca 2016r.

Spis treści

I.	Wstęp.....	3
II.	Opis projektu	3
III.	Wykorzystane technologie	3
IV.	Wykorzystany sprzęt	4
V.	Wymagania systemowe	5
VI.	Implementacja – kluczowe fragmenty.....	5
VII.	Instrukcja obsługi	8
VIII.	Zdjęcia z działania programu	8
IX.	Uwagi	11
X.	Źródła	11

I. Wstęp

Celem projektu jest stworzenie oprogramowania i odpowiednie skonfigurowanie sprzętu do przechwytywania obrazu z kamery, następnie odczytywanie przez komputer obrazu i przekazywanie odpowiednich poleceń do innej aplikacji – w tym przypadku gry komputerowej.

II. Opis projektu

Wstępne założenia projektu zakładały zaimplementowanie następujących funkcjonalności:

- Działanie na systemie Windows
- Obsługa wszystkich zainstalowanych kamer
- Obsługa wszystkich zainstalowanych ewentualnych kart video / grabber-ów
- Wyświetlanie odpowiednich okien z widokiem „na żywo” przechwytywania obrazu
- Możliwość trybu gry z komputerem jak i innym graczem „na żywo”
- Monitorowanie wszystkich zdarzeń (logi aplikacji)

Interfejs aplikacji zajmuje praktycznie cały ekran, składa się z trzech okien, dwa okna dotyczą przechwytywania obrazu „na żywo”, trzecie okno jest to nasza gra „kółko i krzyżyk” do którego polecenia przekazywane są poprzez przechwytywanie obrazu z kamery.

III. Wykorzystane technologie

- Visual Studio 2015
- OpenCV 3.1
- Język programowania C++

Powyższe technologie wybrałem ze względu na znajomość środowiska Visual Studio i języka C++. OpenCV posiada już skompilowane biblioteki, które można dołączyć do projektu w Visual Studio co w znaczący sposób ułatwiło i przyspieszyło pracę. Dodatkowo biblioteki te pozwoliły na zrealizowanie potrzebnych nam funkcjonalności takich jak:

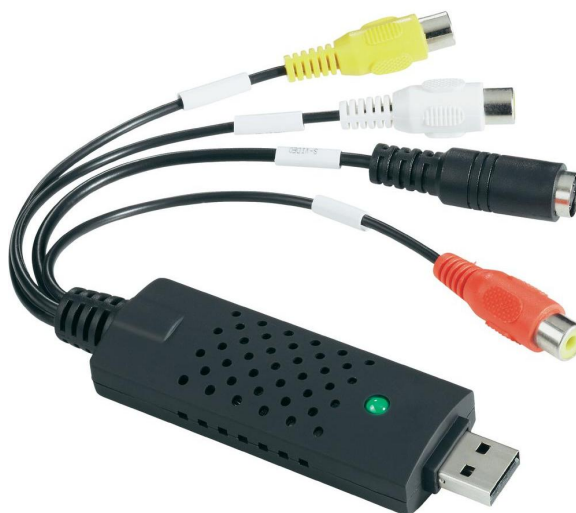
- przechwytywanie obrazu z kamery,
- przetwarzanie przechwyconego obrazu,
- narzędzia do śledzenia obiektów w obrazie,
- stworzenie prostego interfejsu graficznego.

IV. Wykorzystany sprzęt

Do wykonania projektu konieczne jest użycie dodatkowego sprzętu, tzn. kamery (może być kamera internetowa zainstalowana w laptopie jednak znacząco utrudni to rozgrywkę, będzie ona praktycznie niemożliwa), dlatego zalecana jest dodatkowa kamera umieszczona na statywie, skierowana pionowo w dół na dobrze oświetlony stół. Ja użyłem kamery cyfrowej Panasonic model HDC-TM60.



Do powyższej kamery dodatkowo musiałem zastosować urządzenie do przechwytywania obrazu video, gdyż kamera nie posiada opcji transmisji obrazu po kablu USB. Tak więc wykorzystałem do tego poniższy grabber:



V. Wymagania systemowe

- System operacyjny Windows
- Do uruchomienia oprogramowania wymagana jest instalacja OpenCV 3.1

VI. Implementacja – kluczowe fragmenty

Przechwytywanie obrazu:

```
int main() {
    cv::VideoCapture capWebcam(0);

    if (capWebcam.isOpened() == false) {
        std::cout << "error: capWebcam not accessed successfully\n\n";
        return(0);
    }

    cv::Mat imgOriginal;
    cv::Mat imgGrayscale;
    cv::Mat imgBlurred;
    cv::Mat imgCanny;

    char charCheckForEscKey = 0;

    while (charCheckForEscKey != 27 && capWebcam.isOpened()) {
        bool bInFrameReadSuccessfully = capWebcam.read(imgOriginal);

        if (!bInFrameReadSuccessfully || imgOriginal.empty()) {
            std::cout << "error: frame not read from webcam\n";
            break;
        }

        cv::cvtColor(imgOriginal, imgGrayscale, CV_BGR2GRAY);

        cv::GaussianBlur(imgGrayscale,
            imgBlurred,
            cv::Size(5, 5),
            1.8);

        cv::Canny(imgBlurred,
            imgCanny,
            50,
            100);

        cv::namedWindow("imgOriginal", CV_WINDOW_NORMAL);
        cv::namedWindow("imgCanny", CV_WINDOW_NORMAL);

        cv::imshow("imgOriginal", imgOriginal);
        cv::imshow("imgCanny", imgCanny);

        charCheckForEscKey = cv::waitKey(1);
    }

    return(0);
}
```

Gra „Kółko i krzyżyk”

```
public void Play()
{
    if (x is HumanPlayer)
    {
        int position;
        String Playername1, Playername2;
        Map map = new Map();

        Console.WriteLine("\n\nPodaj imię pierwszego gracza (X): ");
        Playername1 = Console.ReadLine();

        Console.WriteLine("\n\nPodaj imię drugiego gracza (O): ");
        Playername2 = Console.ReadLine();

        HumanPlayer player1 = new HumanPlayer('X', Playername1);
        HumanPlayer player2 = new HumanPlayer('O', Playername2);

        Console.Clear();
        DrawMap(map);

        do
        {
            position = player1.getField(map);
            map.PutSign(position, player1.getSign());
            DrawMap(map);
            if (map.CheckWin(player1.getSign()))
            {
                show.winMessage(player1.getName());
                break;
            }
            if (map.CheckDraw())
            {
                show.DrawMessage();
                break;
            }
            Console.Clear();
            DrawMap(map);
            position = player2.getField(map);
            map.PutSign(position, player2.getSign());
            Console.Clear();
            DrawMap(map);
            if (map.CheckWin(player2.getSign()))
            {
                show.winMessage(player2.getName());
                break;
            }
            if (map.CheckDraw())
            {
                show.DrawMessage();
                break;
            }
        } while (true);
    }

    else
    {
        if (x is ComputerPlayer)
```

Tryb gry z komputerem, implementacja ruchów komputera:

```
6 references
public override int getField(Map map)
{
    Random generate = new Random();
    int position;

    do
    {
        position = generate.Next(0, 8);

    } while (map.getArrayMap[position] == 'X' || map.getArrayMap[position] == 'O');

    return position;
}
```

Implementacja wirtualnej planszy

```
public bool CheckWin(char sign)
{
    // Sprawdza wiersze
    for (int i = 0; i <= 6; i += 3)
    {
        if (ArrayMap[i] == sign && ArrayMap[i + 1] == sign && ArrayMap[i + 2] == sign)
            return true;
    }

    // Sprawdza kolumny
    for (int i = 0; i <= 2; i++)
    {
        if (ArrayMap[i] == sign && ArrayMap[i + 3] == sign && ArrayMap[i + 6] == sign)
            return true;
    }

    // Sprawdza przekątną 0-4-8
    if (ArrayMap[0] == sign && ArrayMap[4] == sign && ArrayMap[8] == sign)
        return true;

    // Sprawdza przekątną 2-4-6
    if (ArrayMap[2] == sign && ArrayMap[4] == sign && ArrayMap[6] == sign)
        return true;

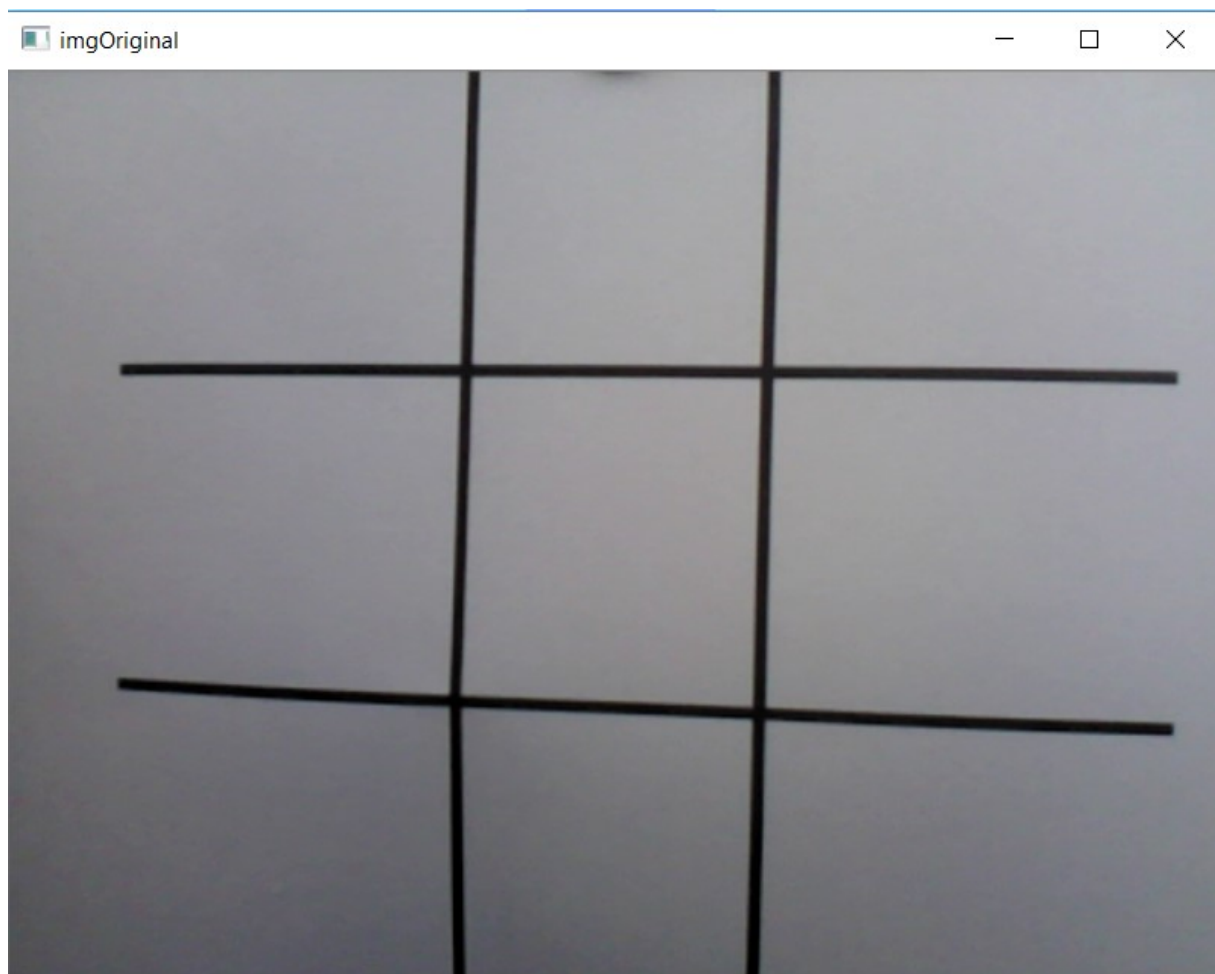
    return false;
}
```

VII. Instrukcja obsługi

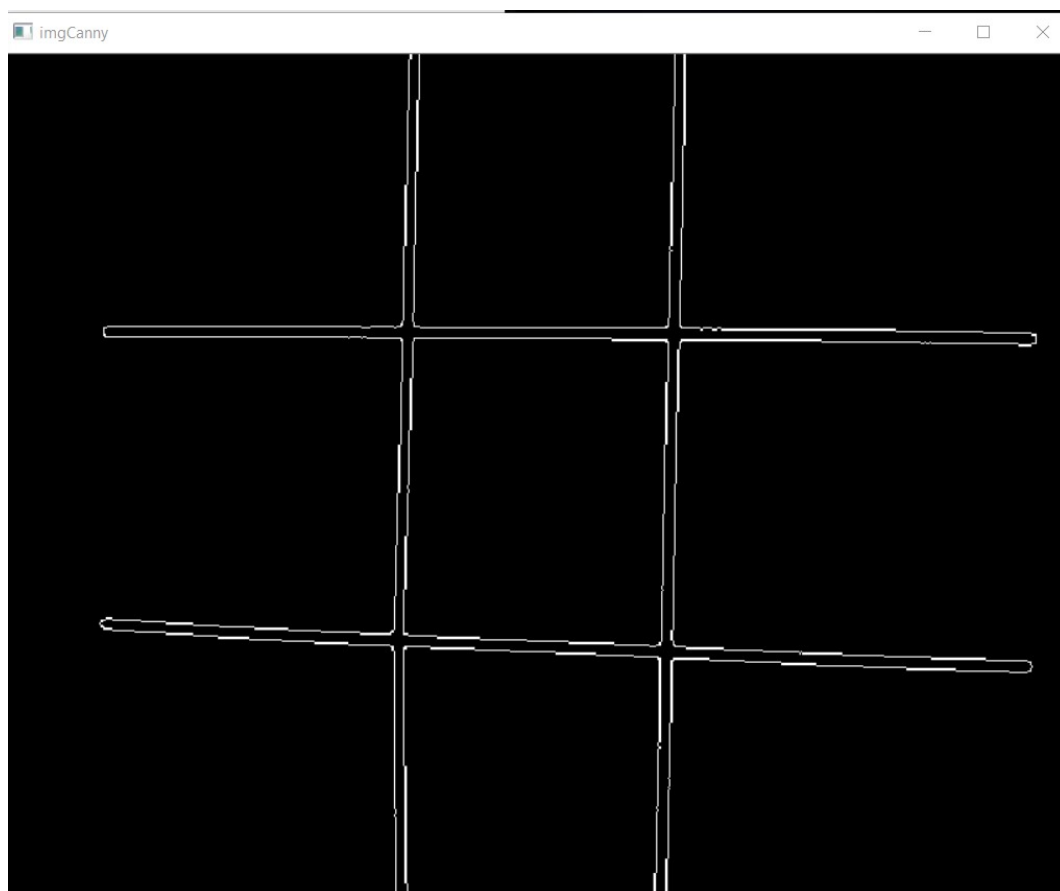
1. Konfigurujemy cały sprzęt
2. Ustawiamy statyw ze skierowaną kamerą najlepiej pionowo w dół na dobrze oświetloną powierzchnię (np. stół)
3. Na stole umieszczamy plansze – kartkę A4 z ramkami do gry w kółko i krzyżyk
4. Uruchamiamy aplikację
5. Wybieramy tryb gry – gra z komputerem, czy z kolegą
6. Ruch 1 przeciwnika – rysuje krzyżyk na planszy i zatwierdza przyciskiem
7. Ruch 2 przeciwnika – jeśli jest tryb z komputerem, to ten punkt jest pomijany, jeśli z człowiekiem to wykonuje on tę samą czynność co 1 przeciwnik, z tym że rysuje kółko i zatwierdza.
8. Taka gra trwa do skutku, do wygranej lub do błędu aplikacji (w przypadku niemożliwości odczytu, niedozwolonych ruchów itp.)

VIII. Zdjęcia z działania programu

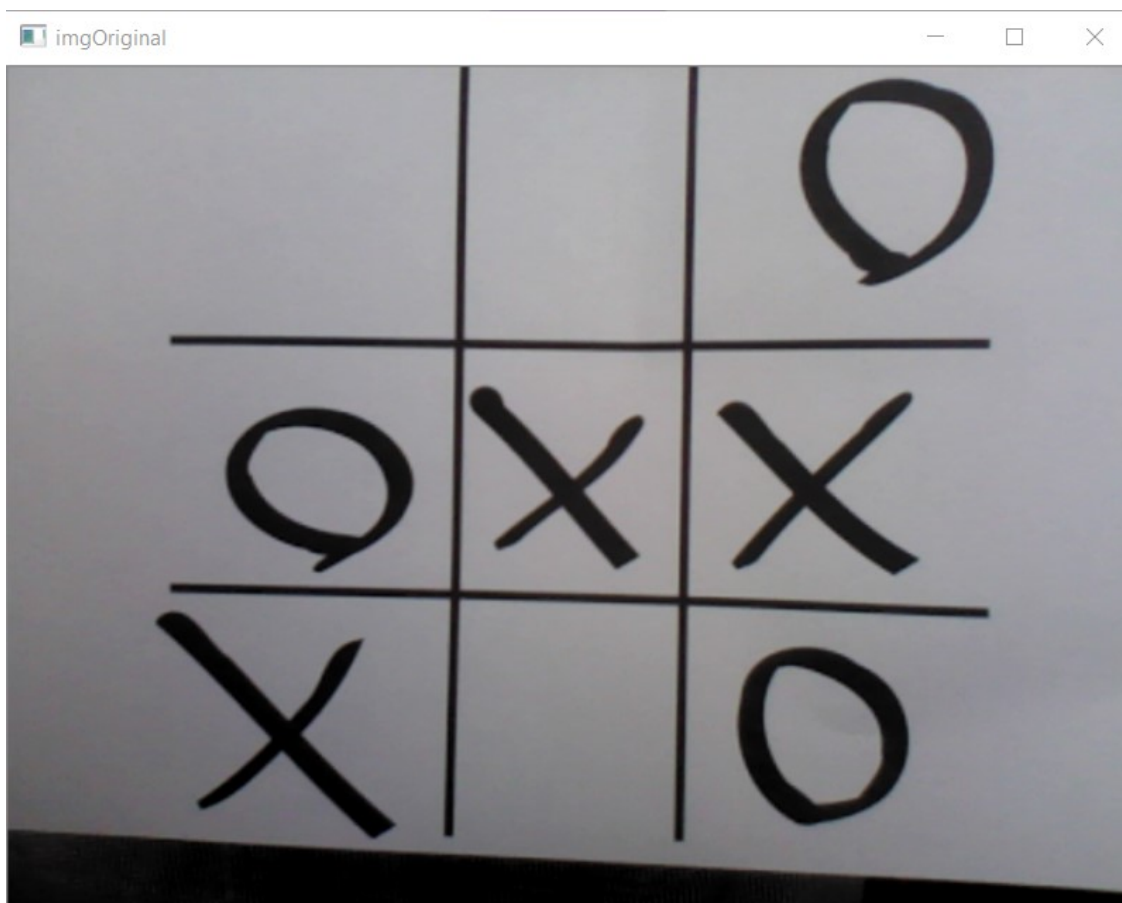
Screen z pierwszego etapu działania aplikacji – wyświetlenie pustej planszy.



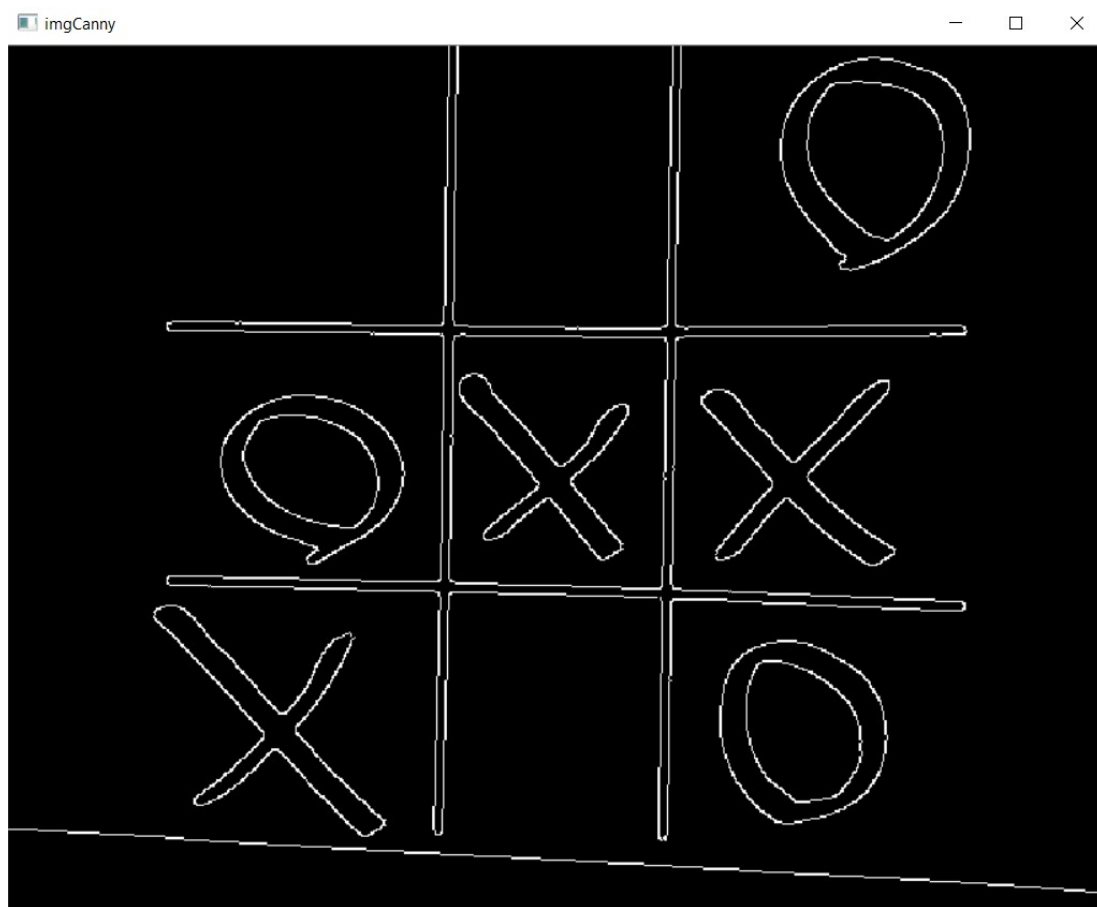
Screen z okna dotyczącego przechwytywania obrazu.



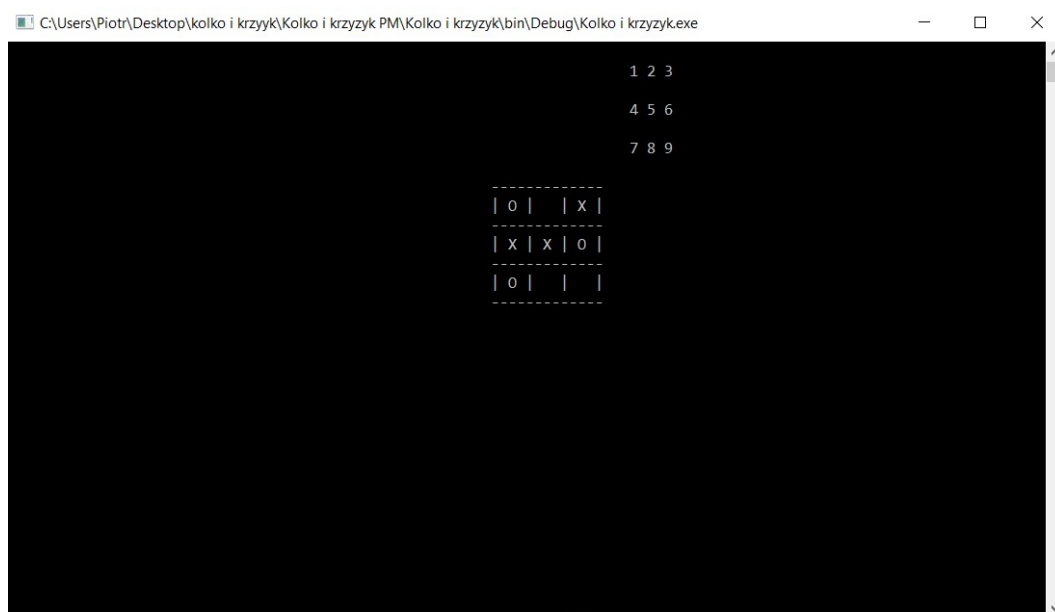
Ponownie widok pierwszego okna już z zarysowaną planszą w trakcie gry



Widok okna z okna odczytywania obrazu



Widok gry „Kółko i krzyżyk”



IX. Uwagi

Projekt został ukończony w takim stopniu jak na finalnej prezentacji. Nie działa on w 100%, pojawiają się problemy z odczytywaniem obrazu i wysyłaniem odpowiednich funkcji do gry, niestety OpenCV jest to dla mnie całkowita nowa technologia, wymagała ona ode mnie dużo pracy i wysiłku, aby opanować chociażby przechwytywanie obrazu. Również z tej racji, że działałem samodzielnie i sam musiałem wyszukiwać wszystkich informacji, to także troszkę utrudniło pracę. Końcowo wykonanie swojego projektu oceniam na 80%.

X. Źródła

www.opencv.org

www.answers.opencv.org

www.cplusplus.com

www.youtube.pl – tutoriale dotyczące opencv

www.google.pl – w przypadku napotykanym problemów wyszukiwanie rozwiązań na różnych forach itp..