

Podstawy teleinformatyki

Rozpoznawanie obrazu z gry w warcaby oraz wizualizacja stanu gry na komputerze

Część III

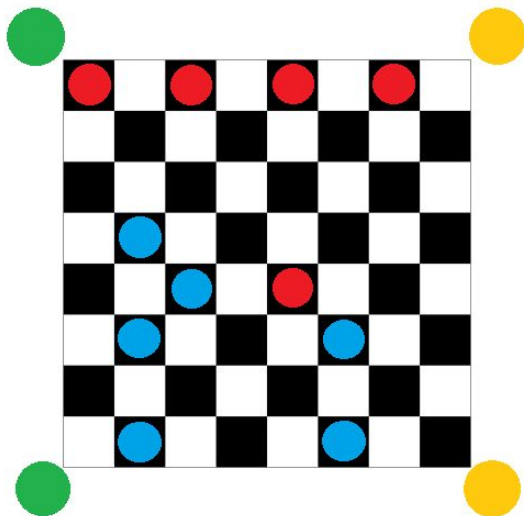
Paulina Mrozek Kornel Krześlak Kamil Sagalara Hubert Springer

Przechwytywanie obrazu z kamery

- jako kamera zostanie użyty telefon z systemem android
- wykorzystana będzie aplikacja "DroidCam"
- na pc musi być aplikacja kliencka
- oba urządzenia muszą pracować w tej samej sieci

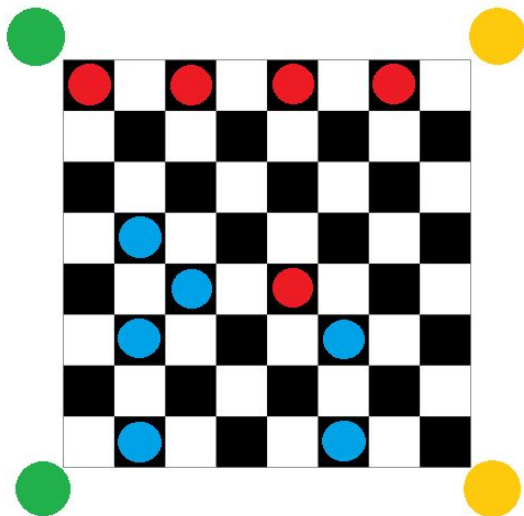


Wykrywanie planszy I - wstęp



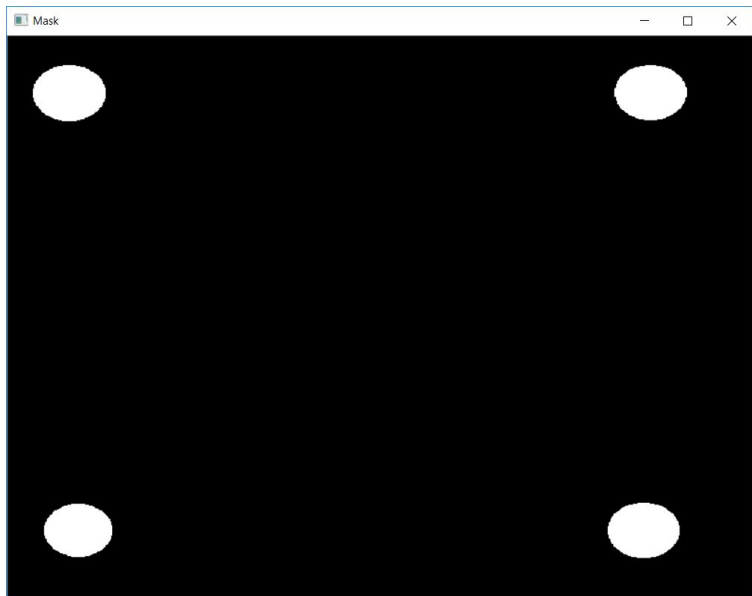
- Umieszczenie markerów o odpowiednim kolorze, można użyć 2 kolorów do określenia lewej i prawej strony
- Odległość markerów jest dowolna, ale im bliżej tym lepiej

Wykrywanie planszy II - barwa



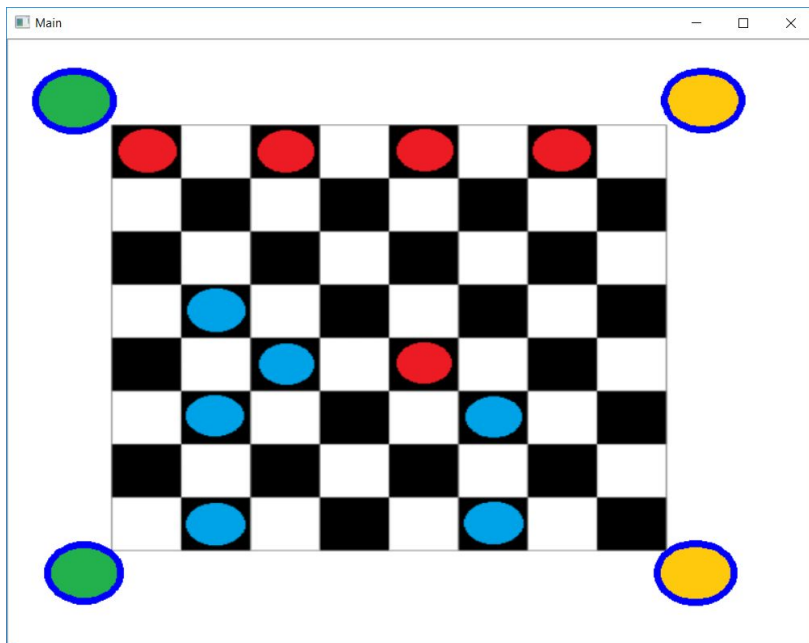
- Przeskalowanie do określonego rozmiaru (np. 800x600)
- Konwersja z przestrzeni barw RGB na HSV
- Ustalenie wartości HSV zgodnie z kolorem markerów

Wykrywanie planszy III - maska



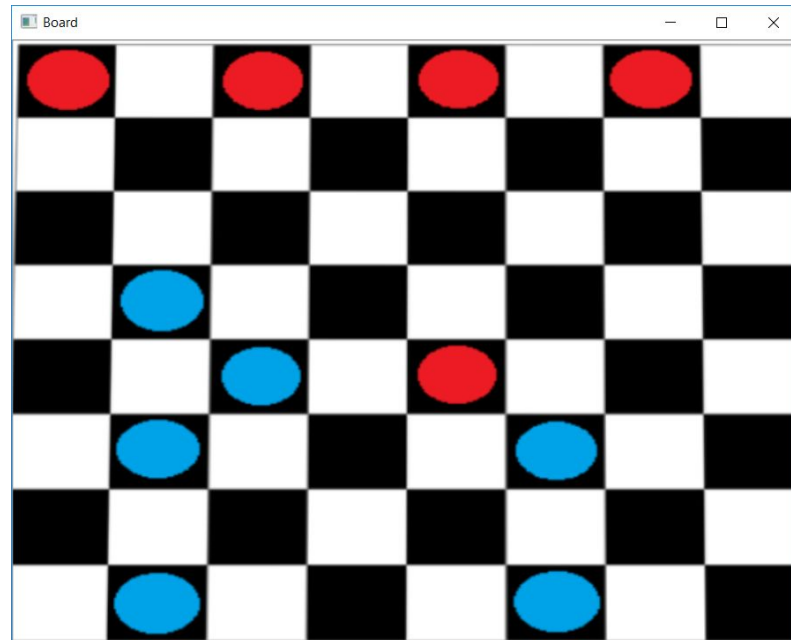
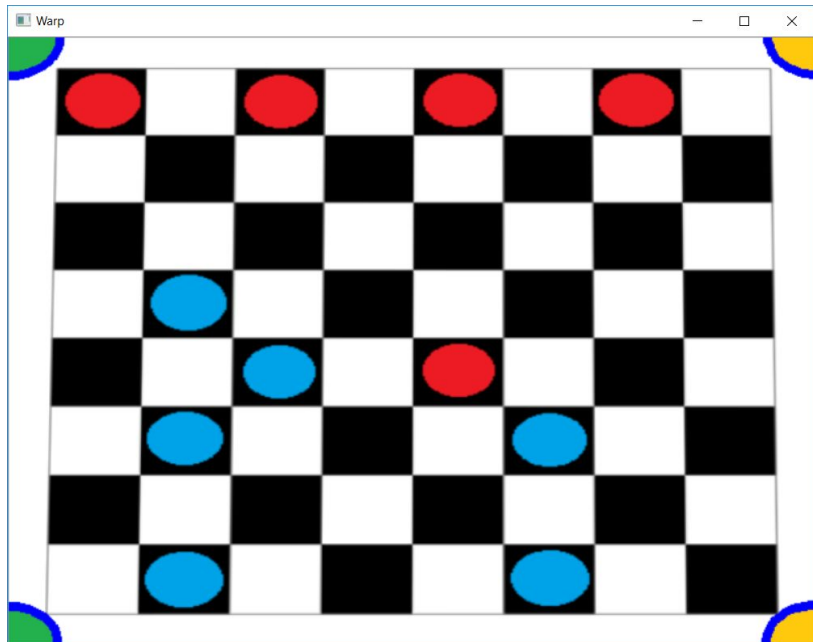
- Wykryte kolory tworzą maskę, na której markery są oznaczone białym kolorem
- Na takim obrazie łatwo jest wykryć potrzebne kontury

Wykrywanie planszy IV - kontury



- Centroid (Środek geometryczny figury) dla każdego markera
- Zastosowanie przekształcenia perspektywicznego (wyrówna obraz w przypadku różnego ustawienia kamery)

Wykrywanie planszy V - wyrównanie obrazu



Wykrywanie planszy VI - podział obrazu

- Podzielenie obrazu na 64 równe bloki i umieszczenie ich na liście
- W następnym kroku każdy blok będzie miał sprawdzany kolor

```
# ustalenie wymiarów planszy
boardHeight, boardWidth, boardChannels = board.shape
# rozmiar pojedynczego bloku
blockHeight = int(boardHeight / 8)
blockWidth = int(boardWidth / 8)

board_blocks = []
for i in range(0, 8):
    for j in range(0, 8):
        board_blocks.append(board[i * blockHeight:i * blockHeight + blockHeight, j * blockWidth:j * blockWidth + blockWidth])
```


Wykrywanie planszy VII - detekcja pionków

- Informacje o każdym polu są przechowywane na liście w postaci liczb całkowitych
- Każdy blok ma sprawdzany kolor środkowego piksela
- Jeżeli wartość odpowiada kolorowi czerwonemu to wartość pola wynosi 2
- Jeżeli wartość odpowiada kolorowi niebieskiemu to wartość pola wynosi 1
- W przeciwnym wypadku wartość pola wynosi 0
- Lista jest przekazywana dalej do podsystemu wizualizacji i poprawności ruchu

Wykrywanie planszy VIII - możliwe trudności

- Nierównomierne oświetlenie może powodować błędy w detekcji planszy i kolorów
- Offset przy wykrywaniu krawędzi musi być odpowiednio dobrany, żeby podział planszy na bloki był wystarczająco precyzyjny
- Na obrazie mogą zostać wykryte niepotrzebne dodatkowe kolory lub kontury, dlatego tło musi być jednolite
- Lista przechowująca kontury może je przechowywać w różnej kolejności, dlatego przed przekształceniem należy określić pozycję każdego z nich (przekształcenie perspektywiczne wymaga podania współrzędnych w odpowiedniej kolejności)

Historia rozgrywki

- kończąc śledzenie ruchów będziemy mieli możliwość zapisania rozegranej partii
 - nadanie własnej nazwy
 - nazwanie graczy
- rozgrywki będą zapisywane w pliku JSON
- będzie możliwość wczytania odbytej rozgrywki i odtworzenia jej
 - automatycznie (stały czas na rundę)
 - ręcznie (przechodząc do kolejnej rundy po naciśnięciu przycisku)

Format zapisu rozgrywki

```
{
  "game_name": "Trial",
  "date": "Jun 1 2018",
  "player1_name": "Andrei",
  "player2_name": "Alex",
  "game_history": [
    {
      "round": "0",
      "pawns": [
        [1,0,1,0,1,0,1,0],
        [0,1,0,1,0,1,0,1],
        [1,0,1,0,1,0,1,0],
        [0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0],
        [0,2,0,2,0,2,0,2],
        [2,0,2,0,2,0,2,0],
        [0,2,0,2,0,2,0,2]
      ]
    }
  ],
}
```

- nazwa rozgrywki
- data rozgrywki
- nazwa (pseudonim) gracza 1
- nazwa (pseudonim) gracza 2
- historia rozgrywki
 - numer rundy
 - pozycje pionków

Sprawdzanie poprawności ruchu

- lista zawierająca wszystkie możliwe dla danego gracza ruchy
- na liście znajdują się tylko najdłuższe sekwencje ruchu
- wykorzystanie rekurencji (wielobicie)
- sprawdzenie czy ruch znajduje się na liście
- przekazanie rezultatu do modułu wizualizacji



