

# Puzzle Matcher-

Aplikacja podpowiadająca ułożenie puzzli

Adam Godziński

Dawid Korach

# Założenia

- Wykrywanie puzzli.
- Układanie na podstawie cech wspólnych.
- Układanie za pomocą kolorów.
- Generowanie obrazu wynikowego.
- Zapisywanie wyniku.

# Technologie

- Język programowania C#.
- EmguCv wrapper do OpenCv.
- Winforms.





# Przebieg Prac



Pracowaliśmy według metodologii scrum. Nasz okres prac podzieliliśmy na 5 Sprintów z czego ostatni był kończącym skupiającym się na debug, dodawaniu brakujących funkcjonalności oraz tworzeniu dokumentacji.



# Przebieg Prac



## Sprint 1

✓ Completed 4

**#2 Opracowanie pomysłu interfejsu**  
Design | Gameplay  3h 1m / 3h



**#5 Gui**  
Design | Gameplay  7h 30m / 8h




**#11 Dodawanie Zdjęć**  
Programming | Feature  1h / 2h

**#12 Creating proper readme file on github (README.md)**  
Programming | Feature  40m / 1h

## Sprint 2




✓ Completed 2




**#8 Seperowanie (usuwanie tła)**  
Programming | Feature  5h / 4h

**#7 Rozpoznawanie/numerowanie puzzli**  
Programming | Feature  10h / 12h

## Sprint 3



✓ Completed 2



**#9 Docelowe umieszczenie puzzli**  
Programming | Feature  22h / 16h



**#13 Układanie puzzli, metoda kolorów - Research**  
Programming | Feature  1h / 1h



## Sprint 4

✓ Completed 4

**#10 Generowanie Obrazu**  
Programming | Feature  4h / 2h

**#15 Dodanie ustawień dla znajdowania puzzli**  
Programming | Feature  4h / 4h

**#16 Dodanie oryginalnego obrazu przez użytkownika**  
Programming | Feature  30m / 1h

**#17 Automatyczne wykrywanie jak mają zostać ułożone puzzle**  
Programming | Feature  4h / 6h

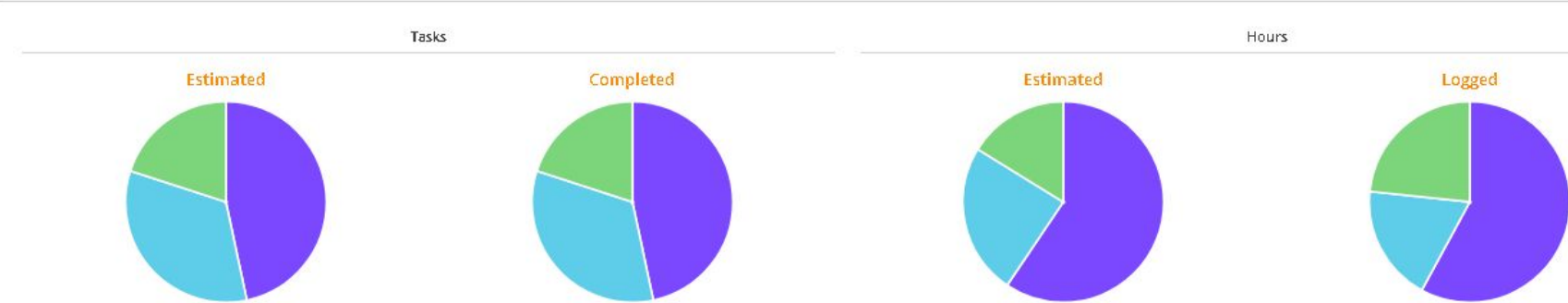
# Przebieg Prac

## Podsumowanie

Category Metrics

	Tasks				Hours		
	Estimated	Pending	Closed	Progress	Estimated	Logged	Unused
Programming	7	0	7	100%	44h	49h 30m	- 5h 30m
Design	5	0	5	100%	18h	16h 1m	1h 59m
Documentation	3	0	3	100%	12h	19h 55m	- 7h 55m
	15	0	15	100%	74h	85h 26m	- 11h 26m

Category Distribution



# Funkcjonalności

- Tworzenie podglądu znalezionych puzzli.
- Układanie puzzli.
- Generowanie obrazu już ułożonych puzzli.

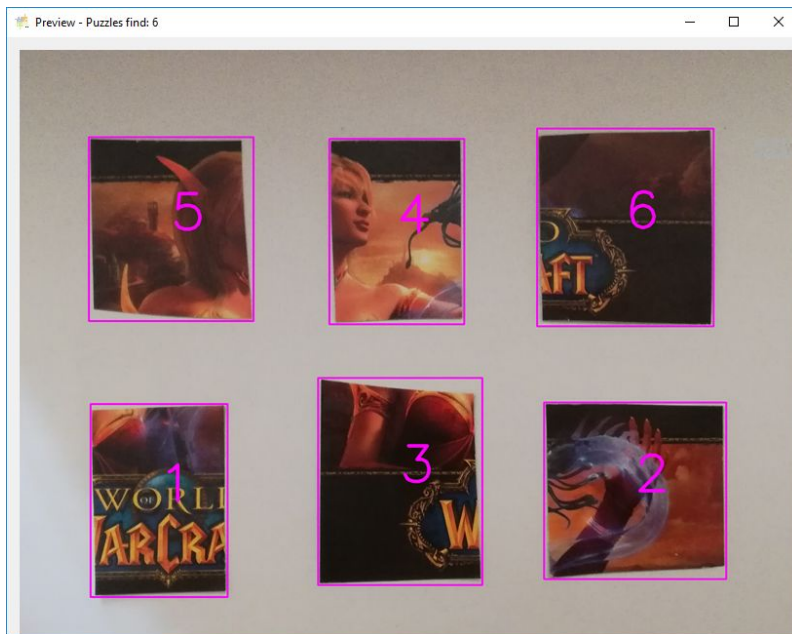
# Funkcjonalności

- Wykrywanie puzzli.

Opiera się ono o wykrywanie konturów oraz o wstępną obróbkę zdjęcia, dzięki którym łatwiej je wykryć.

Program tworzy podgląd znalezionych puzzli.

Dzięki temu, użytkownik jest w stanie sprawdzić co zostanie ułożone.





# Funkcjonalności

- Układanie puzzli.

Puzzle są dopasowywane na podstawie wykrytych cech (SURF).

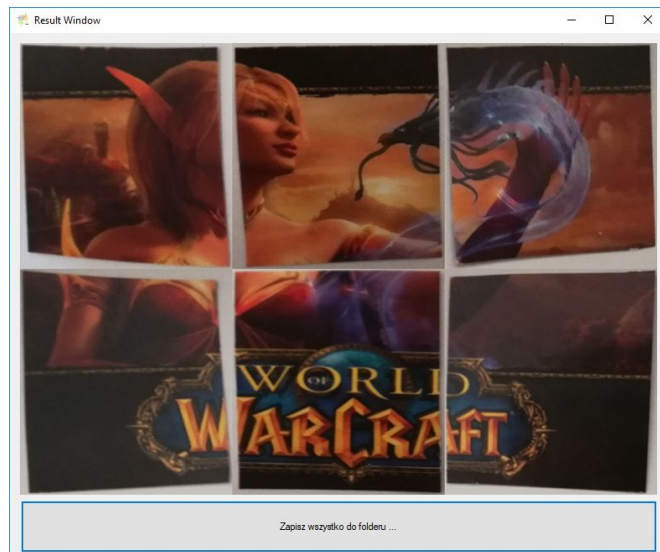
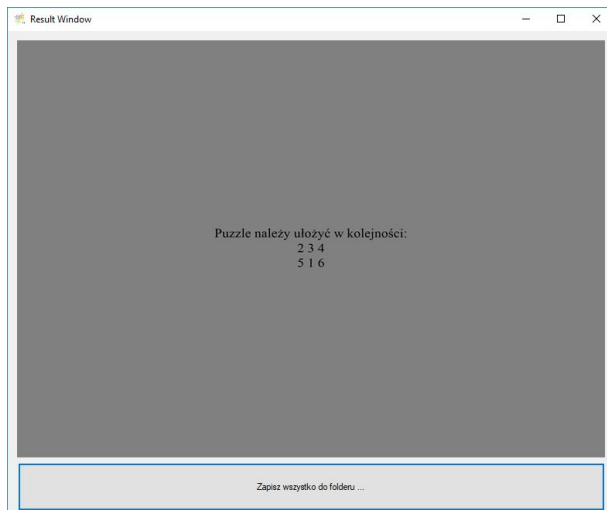
Następnie cechy są dopasowywane względem ułożonego obrazka.

Następnie na podstawie wykrytych cech wspólnych liczymy średnie pozycje puzzli względem oryginalnego obrazu i układamy na podstawie wykrytej/zadanej konfiguracji.



# Funkcjonalności

- Generowanie obrazu ułożenia układanki.  
Na początku wyliczana jest średnia wielkość puzzla do której skalowane są wszystkie puzzle. Następnie puzzle są dodawane po kolei według wyliczonej konfiguracji do obrazu wynikowego.



Prezentacja działania :



# Podsumowanie

- Zaimplementowana przez nas aplikacja jest w stanie wykrywać puzzle.
- Układa puzzle na podstawie ułożonego obrazka.
- Niestety nie za każdym razem obraz jest układany poprawnie, dużo zależy od zdjęcia wejściowego i “stopnia trudności” puzzli.
- Nie został zaimplementowany przez nas algorytm układania puzzli po kolorach który mieliśmy w planach.
- Predykcja konfiguracji ułożenia puzzli nie zawsze działa.

