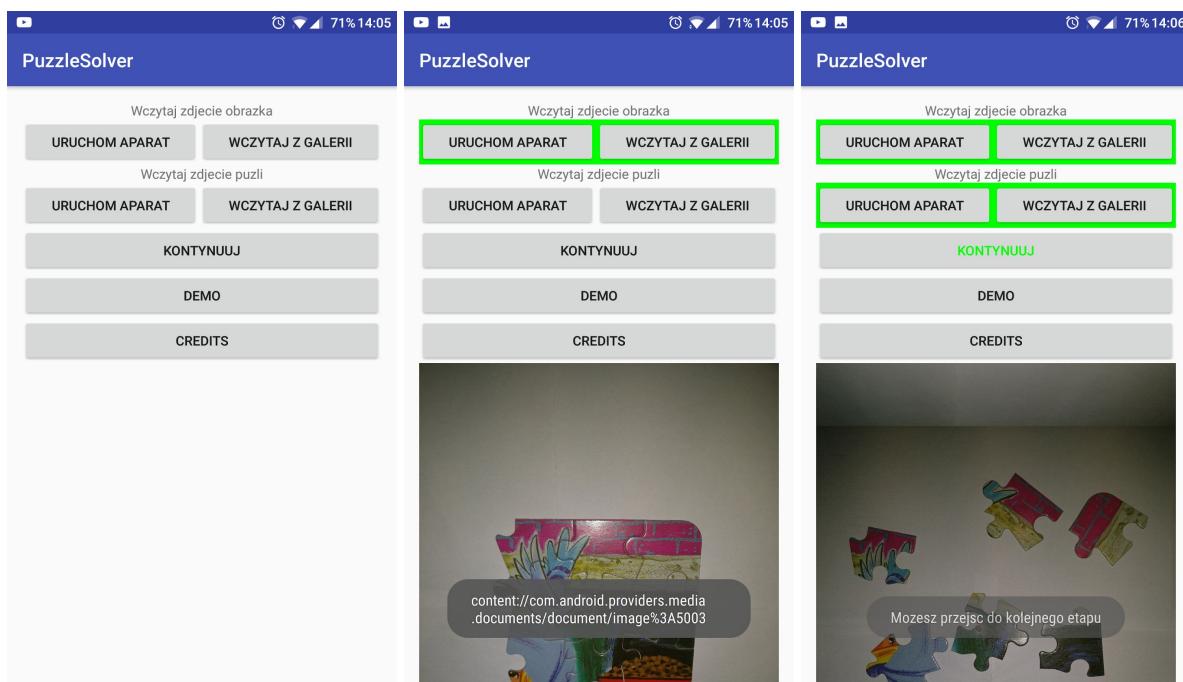


Imię i nazwisko: Mateusz Linke	Nr indeksu: 119399	Rok: 4	Semestr: VII	Wydział: Elektryczny	Rok Akademicki 2016/2017
Data: 27.01.2017	Przedmiot: Podstawy teleinformatyki			Temat: Podpowiadarka do puzzli	

1. Założenia projektu:

- Aplikacja na platformę Android
- Wykorzystanie biblioteki OpenCV
- Po zrobieniu zdjęcia puzzli, aplikacja podpowiada w jaki sposób je ułożyć

2. Interfejs programu:

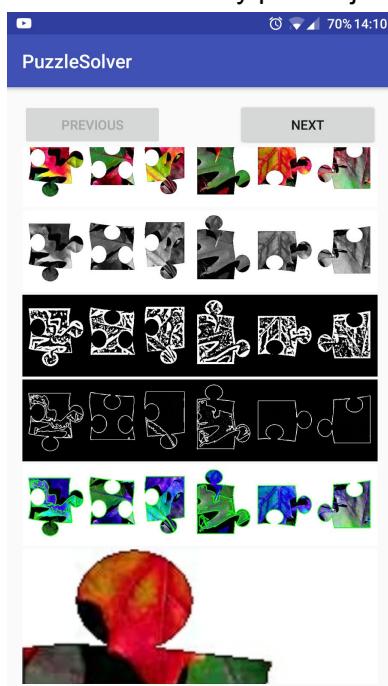


Główny interfejs aplikacji pozwala nam na wczytanie zdjęć nad którymi chcemy pracować. Etykiety "Wczytaj zdjęcie obrazka" oraz "Wczytaj zdjęcie puzzli" sugerują które przyciski są odpowiedzialne za wczytanie poszczególnych zdjęć. Zdjęcia mogą być wczytane bezpośrednio z pamięci telefonu "Wczytaj z galerii" - uruchamia podstawową przeglądarkę do zdjęć, lub z aparatu "Uruchom aparat" - uruchamia aplikacje standardowego aparatu, po zrobieniu zdjęcia przekazuje je do aplikacji. Zielona ramka, która pojawia się po wczytaniu zdjęć informuje, które zdjęcie zostało wczytane. W dolnej części aplikacji możemy podejrzeć zdjęcia. Jest możliwość ponownego wybrania obrazów. Jeżeli zaakceptujemy wybór, klikamy "Kontynuuj" - przechodzimy do kolejnego ekranu odpowiedzialnego za przetwarzanie zadanych zdjęć.

Przycisk "Demo" służy do wczytania zdjęć demonstracyjnych.

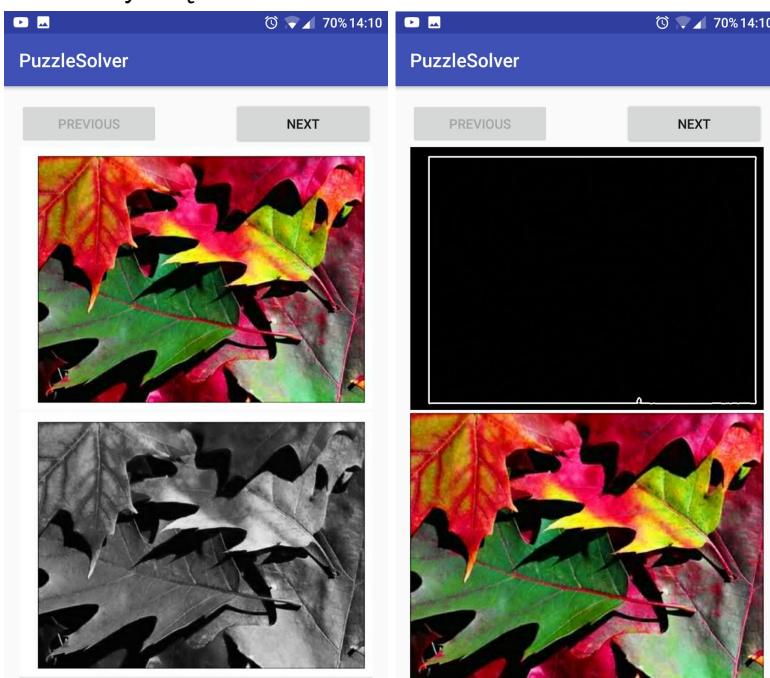
3. Wyodrębnienie puzzli ze zdjęcia.

- Wczytanie oryginalnego zdjęcia, ewentualne zmniejszenie rozdzielczości jeżeli wynosi powyżej 4k
- Konwersja na obraz czarno-biały
- Progowanie adaptacyjne - w celu “odcięcia” elementów od tła
- Znalezienie konturów, następnie filtrowanie, aby odrzucić te błędne(zbyt duże i zbyt małe)
- Nanieśnięcie konturów na obraz
- Wyodrębnienie poszczególnych puzzli tak aby każdy był osobnym elementem listy (na zrzucie ekranu jest pokazany jeden przykładowy puzzle)
- Efek: każdy puzzle jest osobnym elementem w liście



4. Wyodrębnienie zdjęcia całego obrazka.

- Wczytanie oryginalnego zdjęcia, ewentualne zmniejszenie rozdzielczości jeżeli wynosi powyżej 4k
- Konwersja na obraz czarno-biały
- Progowanie adaptacyjne - w celu "odcięcia" elementów od tła
- Znalezienie konturów, następnie filtrowanie, aby odrzucić te błędne(znaleźć jeden, największy, ale mniejszy niż cały obraz)
- Nanieśenie kontury na obraz
- Wyodrębnienie obrazu bez ramek z tła



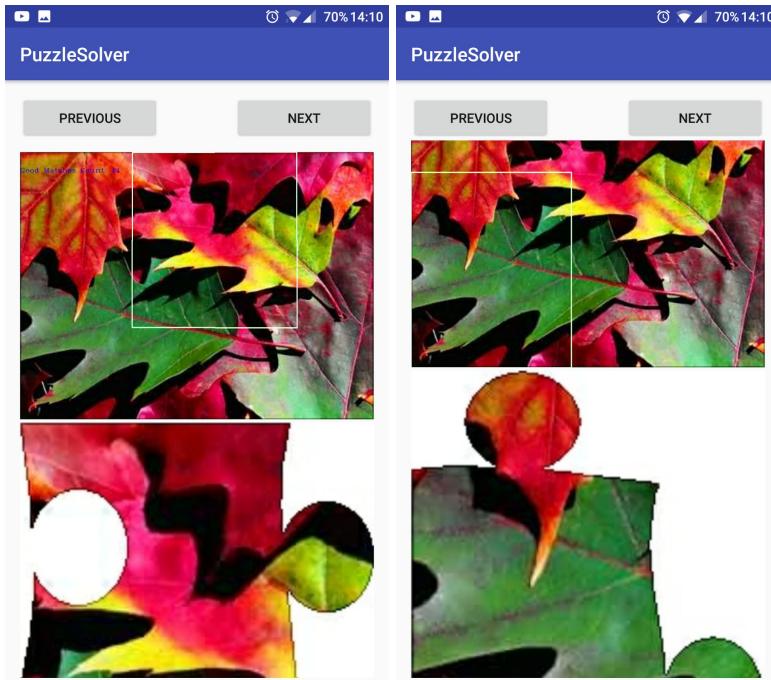
5. Znajdowanie cech szczególnych(krawędzi) puzzli i pełnego obrazu.

- Przed przystąpieniem do szukania cech należy obraz zmienić na monochromatyczny(czarno-biały)
- Następnie za pomocą algorytmu FAST(Features from Accelerated Segment Test) szukamy cech (zaznaczone jako kolorowe kółka). Jak widać problemem jest znajdowanie punktów na krawędziach pojedynczego puzzla, niestety nie udało mi się wyeliminować tego problemu.



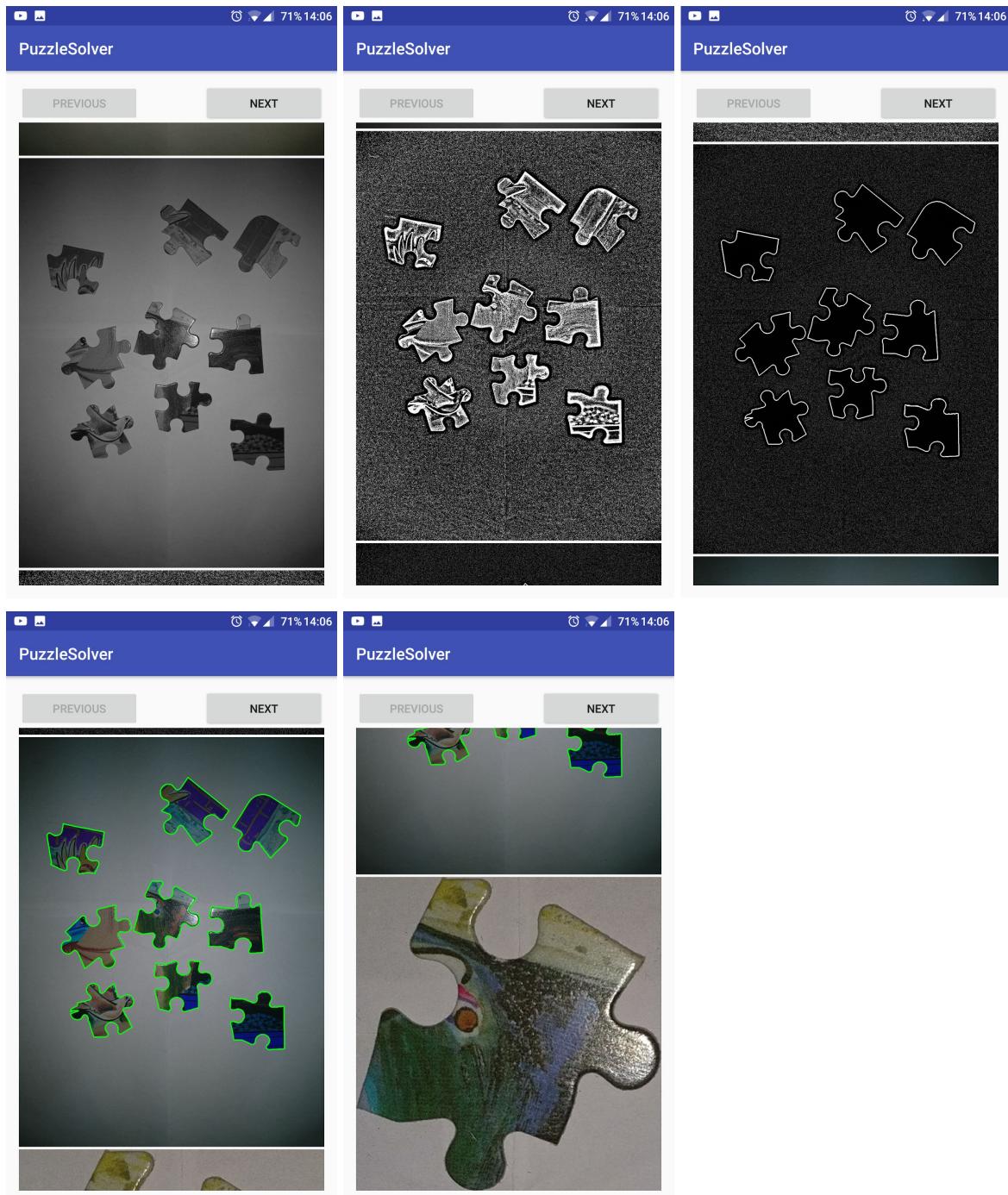
6. Znajdowanie puzzla na obrazie.

- Porównujemy znalezione cechy za pomocą algorytmu ORB (Oriented FAST and Rotated BRIEF), algorytm jest fuzją znanych wcześniej algorytmów(niektóre są opatentowane), stworzony przez “OpenCV Labs”
- Filtrujemy wyniki aby otrzymać tylko “dobre połączenia”
- Na zrzutach ekranu widać jakie miejsca przypisał algorytm danym puzzlom

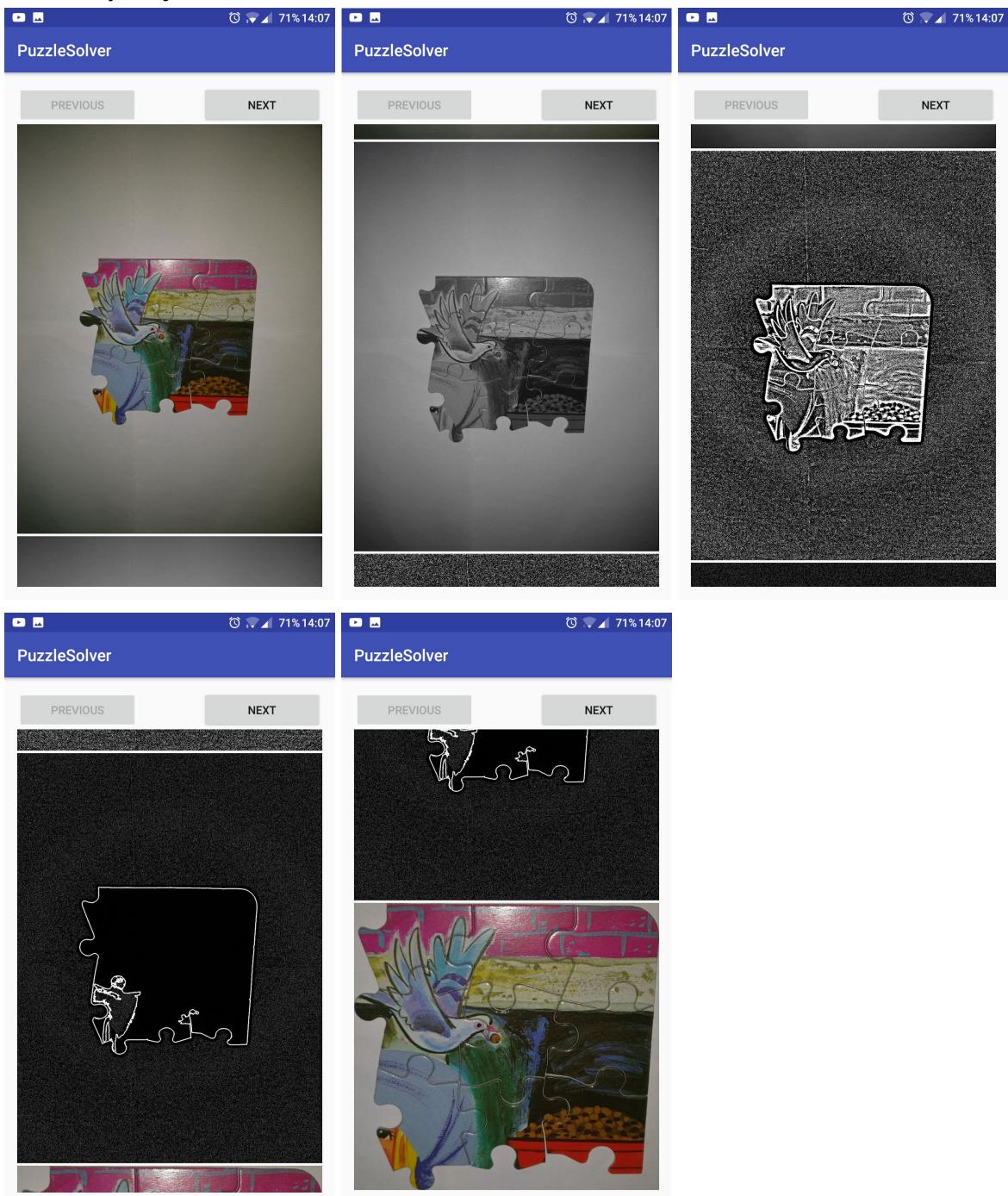


7. Wyniki otrzymane na realnych zdjęciach.

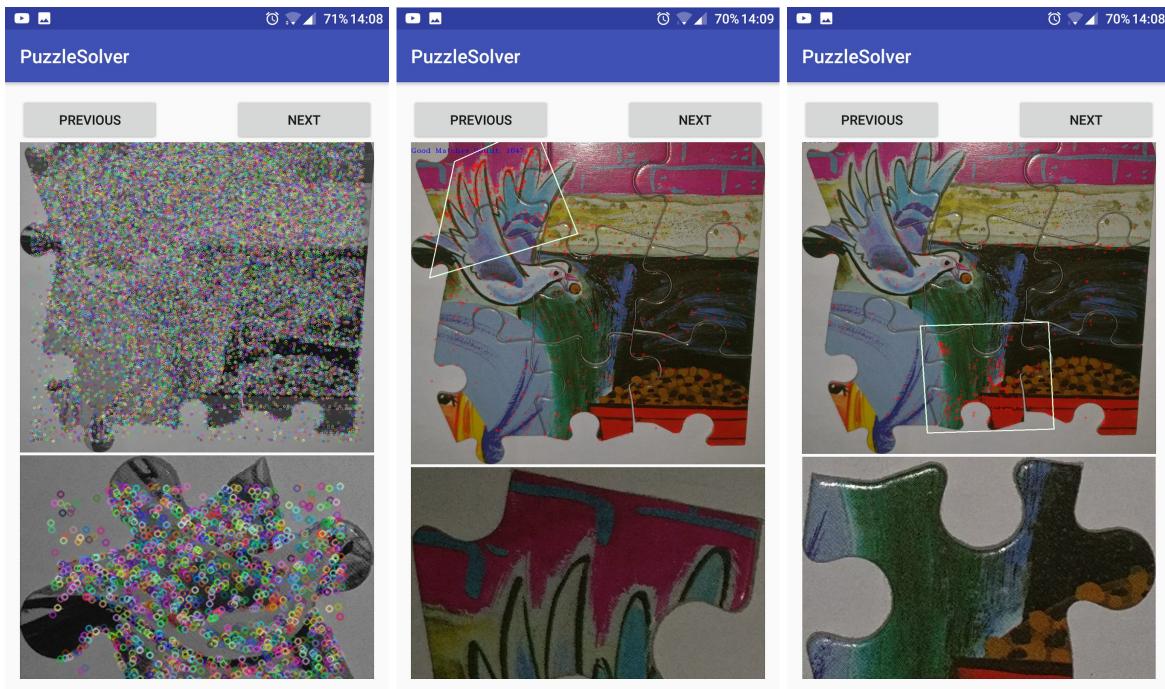
- Procedura przebiega tak samo jak dla powyższych zdjęć testowych.
- Wyodrębnienie puzzli



- Wyodrębnienie obrazu



- Znalezienie puzzli na obrazie



8. Znane problemy.

- Oświetlenie ma duży wpływ na wyniki.
- Błędne wyniki na niektórych urządzeniach, prawdopodobnie wynika to z implementacji biblioteki OpenCV
- Ograniczona liczba algorytmów dostępnych w Javie oraz problematyczne zmienianie parametrów tych algorytmów.
- Do znajdywania krawędzi na zdjęciu są dostępne tylko 2 (dobre)algorytmy:
 - FAST - dosyć szybki lecz niedokładny, wykryte krawędzie mają małą wiarygodność i sporą część trzeba odfiltrować
 - ORB - przy ustawieniach standardowych znajdował bardzo małą ilość krawędzi

9. Podsumowanie.

Projekt wymagał zaznajomienia się z wieloma, wcześniej nieznanymi mi pojęciami oraz algorytmami przetwarzania obrazu. Sama biblioteka OpenCV też nie jest idealna, wiele funkcji nie działa w Javie i trzeba robić to "na około". Korzystanie z oficjalnej dokumentacji również nie należało do najprzyjemniejszych, cały kod i przykłady są napisane w C++ lub Pythonie co stwarzało pewne problemy i brak kompatybilności z wrapperem Javy. W rezultacie jednak, po wielu nieprzespanych nocach udało się osiągnąć zadowalający efekt.

8. Link do repozytorium.

<https://github.com/anuluj/PuzzleSolver>