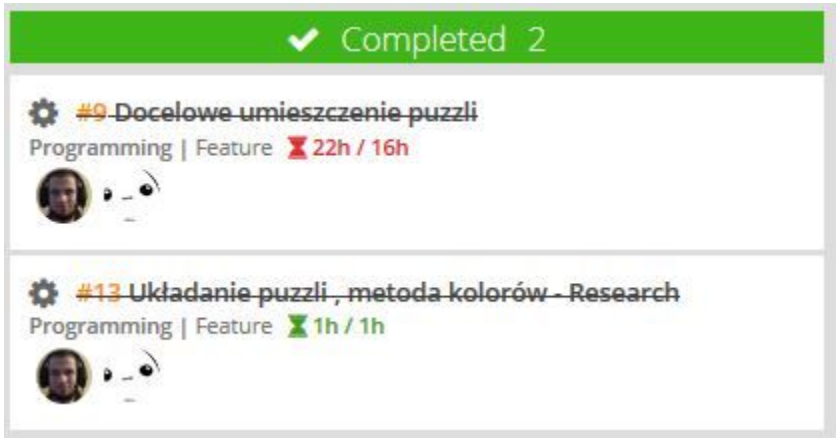
Wykonawcy:	Index:	Rok: 3; Semestr: 5	Podstawy Teleinformatyki (26.04.2018)
Adam Godziński	127065	Język programowania:	C#
Dawid Korach	126909	Temat:	Podpowiadarka do puzzli

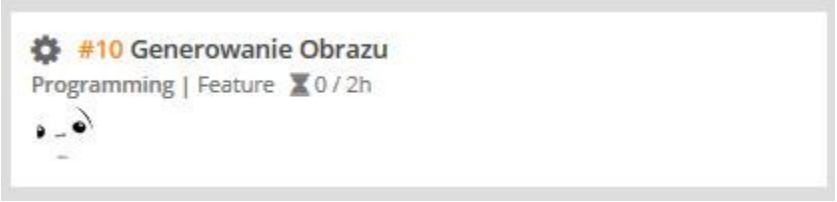
Sprint 3



Celem tego sprintu było ułożenie puzzli przy pomocy key-pointów oraz wygenerowanie obrazu złożonych puzzli.

Zadania związane z dopasowywaniem puzzli okazało się trudniejsze niż zakładaliśmy , ale koniec końców udało nam się to zrobić.

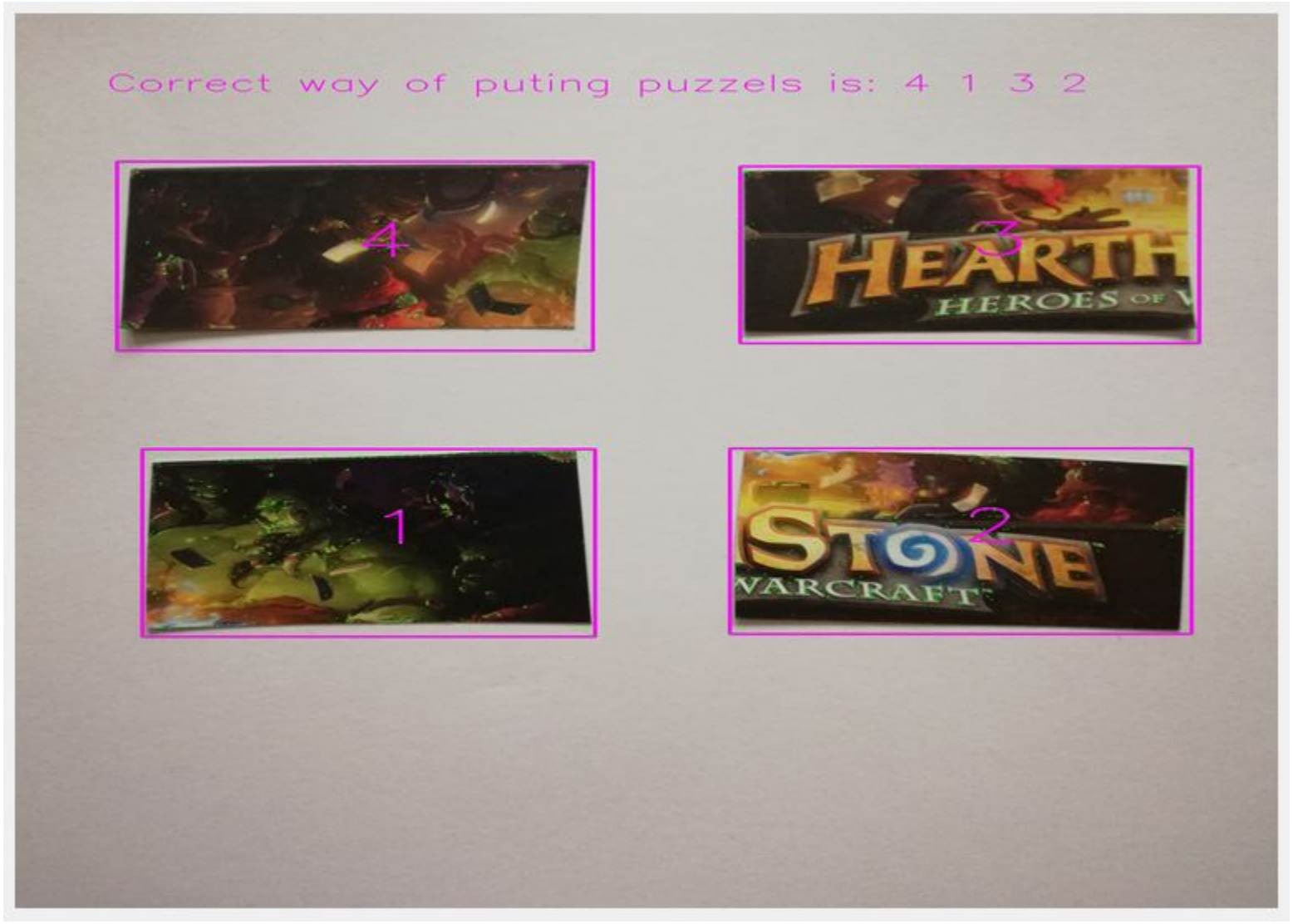
Nie udało nam się jednak zaimplementować generowania ostatecznego obrazu, więc mamy zamiar przenieść to zadanie na następny Sprint.



Obraz oryginalny.

Zdjęcia użyte do prób.

Dopasowane key-point’y.



Jeden z obrazów wynikowych programu.

Podsumowanie

Podczas testów największe problemy mieliśmy z liczebności key-point'ów.
Przez długi czas, algorytm, który napisaliśmy, znajdował ich za mało. Ostatecznie, udało się nam dobrać parametry wystarczające, aby program poprawnie dobierał puzzle do siebie.

Oprócz tego, do programu został dodany wymóg podania zdjęcia z już ułożonymi puzzlami.
Na tym właśnie zdjęciu, opiera się nasz rdzeń algorytmu

```
Features2DToolbox.DrawKeypoints(orginal, orginalKeypoints, copyOrginal, new Bgr(0, 255, 0));
```

.

Sam algorytm, oprócz tego, wymaga jeszcze podania rozmiaru ułożonych puzzli - tzn. podania jego wysokości i szerokości w jednostce ilości puzzli.

```
.placePuzzels(2, 2, avgPuzellXPoints, avgPuzellYPoints);
```

Fragment wywołania metody zawierającej algorytm. W tym wypadku mówimy algorytmowi, że ułożony obraz ma 2 puzzle szerokości i 2 puzzle wysokości.

Ponieważ staraliśmy się, aby w pierwszej kolejności kod działał, to w kolejnym sprincie będziemy musieli poprawić jego wydajność (refaktoryzacja i rewizja).