**Zawansowane Języki Programowania**

**Projekt  
refaktoryzacja kodu**

**Aleksander Kosma | 238193  
Dawid Ćwik | 238137  
15.01.2019.r**

## Opis projektu Wykonany przez nas projekt był rozbudowaną wersją symulacji mrówek. W naszej wersji istnieje możliwość wybrania ilości mrówek biorącej udział w zdobywaniu jak największej ilości powierzchni. Procentowy udział każdej z mrówek jest wyświetlany wraz z mapą. Na mapie każda mrówka ma swój wlasny kolor, co pozwala odróżniać je od siebie. Poniżej screen z gry:

## 

## Problem refaktoryzacji kodu zaczęliśmy od stworzenia projektu, który jak najniższym kosztem czasowym spełniałby wymagania postawione na początku. Proces ten zabrał jeden wieczór kodowania. Mialo to na celu wykonanie kodu o niskiej jakości, który byłby idealny do rozpoczęcia procesu refaktoringu.

## Wykorzystane narzedzia / technologie By ulatwić sobie szybką diagnoze i lokalizacje blędów, ostrzeżeń czy smell codes, wykorzystaliśmy następujące wtyczki, możliwe do wykorzystania w IDE Intellij Community:

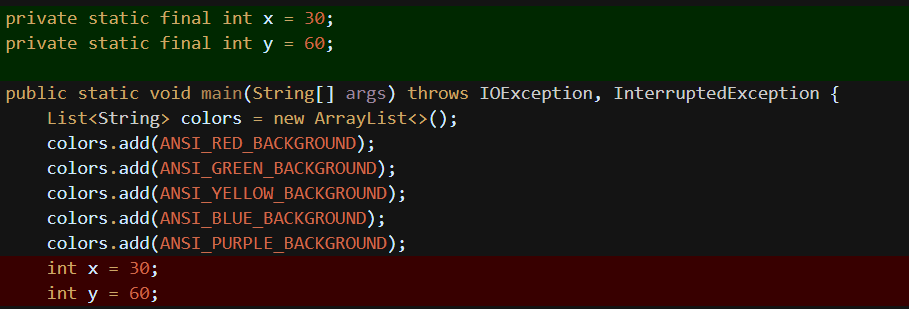
## SonarLint

* CheckStyle – IDEA
* QAPlugin

Dodatkowo korzystaliśmy z:

* serwisu BetterCodeHub
* książki Clean Code by Robert C. Martin

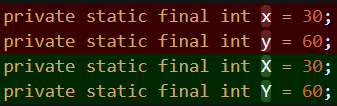
## Kroki refaktoryzacji kodu

Poniżej będą prezentowane kolejne kroki przemiany kodu  
  


* squid : S00115

Sharing some naming conventions is a key point to make it possible for a team to efficiently collaborate. This rule allows to check that all constant names match a provided regular expression.

The following code snippet illustrates this rule when the regular expression value is "^[A-Z][A-Z0-9]\*(\_[A-Z0-9]+)\*$":



* squid : S1125

Remove literal boolean values from conditional expressions to improve readability. Anything that can be tested for equality with a boolean value must itself be a boolean value, and boolean values can be tested atomicly.



* squid:S1659

# Multiple variables should not be declared on the same line.

