# Sprawozdanie z projektu

# na Informatyczne Narzędzia Pracy Grupowej

**SAPER**



Joanna Stanisz

Patrycja Tokarczyk

Kamil Barczyński

Piotr Gołyźniak

## Cel projektu:

* Nauka metodyki Scrum w pracy grupowej.
* Nauka informatycznych narzędzi pracy grupowej takich jak: github, tortoise git.
* Wdrożenie się w z zasady i organizację pracy grupowej.
* Implementacja konsolowej wersji gry *Saper* w języku *C++*.

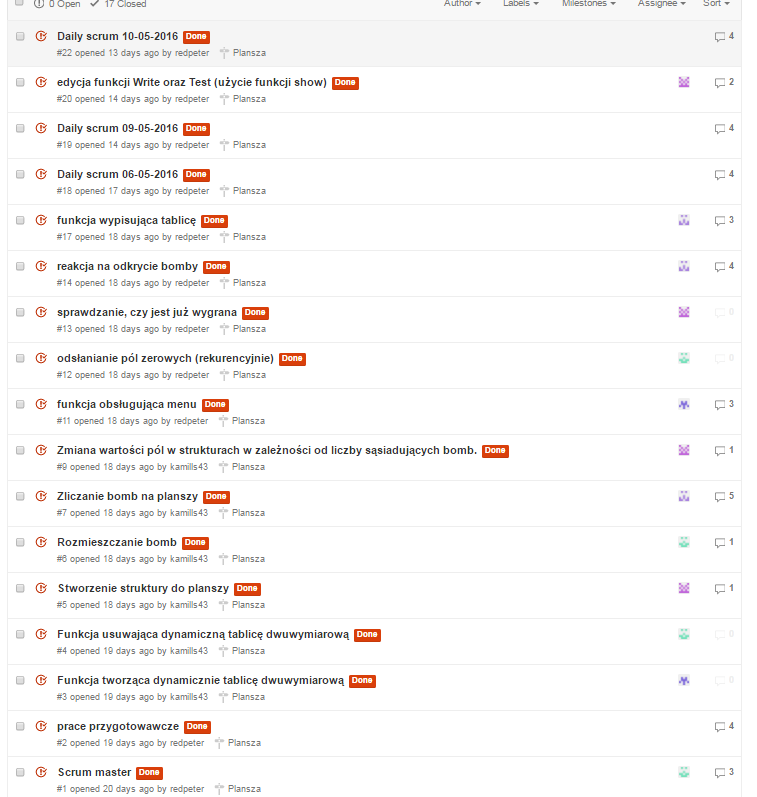
## Saper, czyli?

* gra komputerowa
* oryginalna nazwa to *Minesweeper*, została napisana w 1981 roku przez Roberta Donnera
* polega na odkrywaniu pól planszy w taki sposób, aby nie trafić na minę
* każde odkryte pole zawiera informację o liczbie min, z którymi sąsiaduje (08)
* nieodkryte pola można oznaczać flagą (*domniemana bomba*), przez co blokowane jest jego odsłonięcie i przypadkowe odkrycie miny

## Opis poszczególnych sprintów:

1.Plansza

Pierwszy sprint został przez nas zaplanowany tak, aby w ramach pięciu dni roboczych napisać kod zawierający prototyp gry. Wyznaczyliśmy pierwszy milestone: „Plansza”, który przypominał nam, że celem ówczesnego sprintu jest stworzenie podstaw, do dalszego rozwoju gry. Wybraliśmy także pierwszego Scrum Mastera: Piotra. Ustaliliśmy, że będziemy korzystać z serwisu github.com w celu stworzenia repozytorium, które umożliwiło nam ogólny dostęp do kodu oraz możliwość jego modyfikacji.



        Pierwszy dzień poświęciliśmy na sprawy organizacyjne. Pobraliśmy na nasze komputery system kontroli wersji: Git, oraz oprogramowanie klienckie TortoiseGit umożliwiające sprawne korzystanie z tego systemu. Tego dnia przeprowadziliśmy także pierwsze commity, które pozwoliły rozstrzygnąć, czy nasze programy działają prawidłowo. Wyznaczyliśmy także, zadania na najbliższe dni: dokładne zapoznanie się z zasadami działania gry saper, oraz napisanie kodu, pozwalającego stworzenie tablicy dwuwymiarowej.

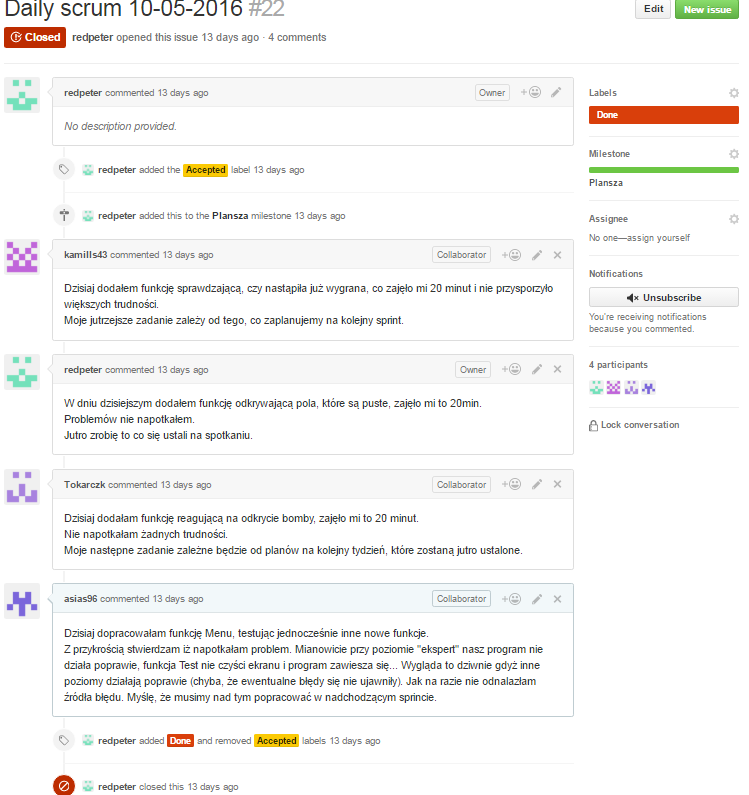
Następnego dnia zabraliśmy się do pisania kodu. Każdy pracował nad przydzieloną mu do napisania funkcją, dlatego już po pierwszym dniu pracy mieliśmy 4 najważniejsze funkcje naszego programu: tworzącą, usuwającą i wypisującą tablice dwuwymiarową oraz strukturę do planszy. Każdy bez problemów wypełnił swoje zadanie a przeprowadzone testy pokazały, że napisane funkcje działają prawidłowo.

Ustaliśmy, że po napisaniu i wprowadzeniu do repozytorium zmian w kodzie będziemy zmienić labels w odpowiednim issue na Test, dzięki czemu będzie wiadomo, że funkcja jest skończona i można zająć się jej testowaniem. Testowanie będzie polegać na sprawdzaniu siebie nawzajem.

Trzeciego dnia rozwijaliśmy stworzone wcześniej podstawy. Dodana została funkcja zliczająca bomby na planszy jednocześnie wyświetlająca ową ilość bomb, funkcja rozmieszczająca losowo bomby, funkcja testowa wyświetlająca tablicę z bombami oraz z ilością bomb pozostałych na planszy. Dokonane zostały też inne niezbędne ulepszenia jak np. dodanie dyrektywy preprocesora dla bomby, zmiany wyświetlania tablicy w funkcji wypisującej czy ujednolicenie kodu.

Czwartego dnia dodaliśmy do naszego programu funkcję odkrywającą pola wybrane przez użytkownika oraz szkielet funkcji obsługującej menu. Zostały dokonane również ulepszenia istniejących już funkcji: Write, Test i Random np. zwiększanie wartości pól wokół bomby. Każdy poradził sobie ze swoim zadaniem nie napotykając po drodze problemów.

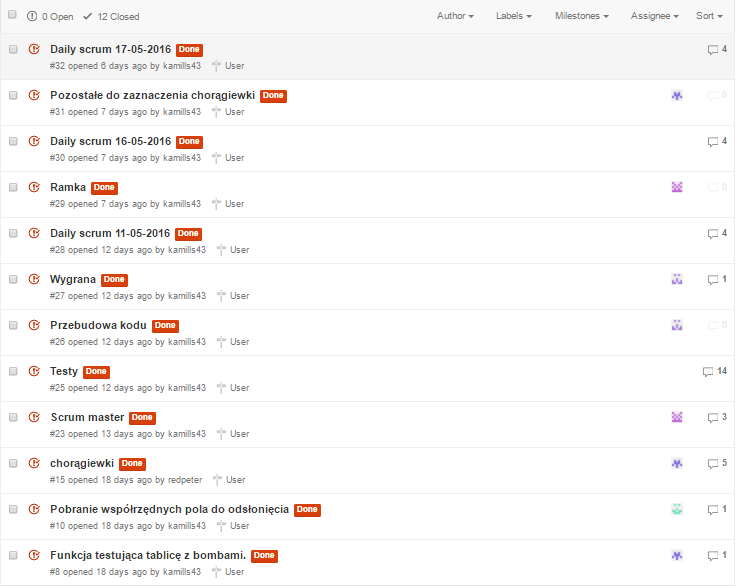
Ostatniego dnia pierwszego sprintu pracowaliśmy nad kolejnymi ulepszeniami zamykającymi milestone “Plansza”. Były to następujące funkcje: sprawdzająca czy nastąpiła już wygrana, reagująca na odkrycie bomby (czyli sygnalizująca przegraną), dokończona funkcja obsługująca menu oraz wykonane rekurencyjnie odsłanianie pól zerowych. Testowanie świeżo dodanych funkcji nie wykazywało błędów, jednak…



 Wykryty został problem, którego źródła nie udało nam się ustalić przed zakończeniem ówczesnego sprintu. Mimo to zamykając milestone “Plansza” mogliśmy pochwalić się sporym nabytkiem napisanego kodu, działającym lecz z malutkim wyjątkiem. Znalezienie powstałego błędu zostało przełożone na kolejny sprint nazwany: “User”.

2.User

W ramach następnych trzech dni roboczych zaplanowaliśmy drugi sprint, którego zakres zawarliśmy w milestonie “User”, wprowadzającym pewne udogodnienia dla użytkownika naszego programu. Wybraliśmy także nowego Scrum Mastera: Kamila.



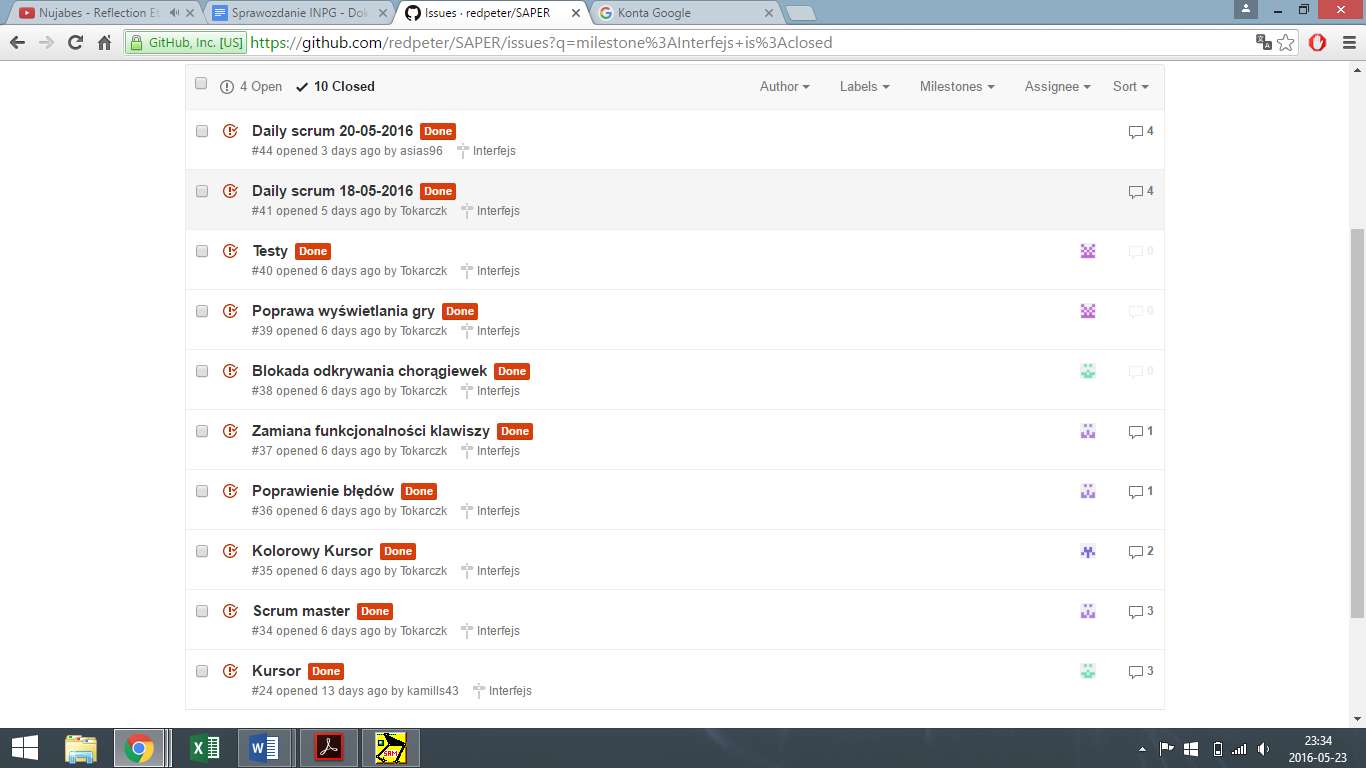
Pierwszego dnia wprowadziliśmy znacznie udogodnienie w użytkowaniu naszego programu, była nim funkcja pozwalająca na poruszanie się przy użyciu *strzałek* po wyznaczonej planszy, a także dostosowanie istniejących już funkcji, do tego przełomowego kroku w naszym projekcie. Ponadto zajęliśmy się poszukiwaniem błędu wykrytego podczas poprzedniego sprintu, który pojawiał się przy tablicach niekwadratowych. Został on wykryty i naprawiony, a cały dotychczasowy kod ujednolicony i uczyniony bardziej czytelnym.

Drugiego dnia wprowadziliśmy nową funkcjonalność w naszej grze, umożliwiającą oznaczenie pola jako flagę. Wiążące się z tym zmiany w funkcjach składowych programu udało się wykonać jeszcze tego samego dnia. Poprawiona została również strona wizualna gry poprzez otoczenie planszy ramką, a także kodu programu, w wyniku dalszego jego uczytelniania. Niestety jeden z członków naszego zespołu napotkał pewne problemy z użytkowaniem biblioteki umożliwiającej nam wszystkie, nowe funkcjonalności, z tego też powodu musiał zmienić swoje środowisko programistyczne, aby móc dalej wspierać nas w tworzeniu naszego produktu.

Ostatniego dnia naszego sprintu umożliwiliśmy użytkownikowi korzystanie z menu naszej gry za pomocą strzałek, a także rozpoczęliśmy żmudny proces optymalizacji kodu oraz tworzenia nowych jego usprawnień. Dzięki flagom oraz zmiennej zakryte mogliśmy zrezygnować z funkcji zliczającej ilość bomb, tym samym zmniejszając złożoność obliczeniową naszego programu.

3. Interfejs

W trzecim sprincie przyłożyliśmy dużą uwagę, na wizualny wygląd naszego programu, co sprawiło, że gra stała się bardziej atrakcyjna. Ustaliliśmy, że przez dwa i pół dnia Scrum Masterem zostanie Patrycja, natomiast przez pozostały czas tę rolę pełnić będzie Joanna.



Pierwszy dzień poświęciliśmy na zebranie informacji na temat możliwości użycia kolorów w naszym programie. Wprowadzone zostały także drobne zmiany w użytkowaniu flag w trakcie gry. Tego dnia podjęliśmy także decyzję o przeprowadzeniu dalszych testów sapera, w celu wyłapania możliwych błędów.

Następnego dnia można już było zobaczyć pełen efekt użycia kolorów, co znacznie poprawiło wygląd naszej gry. Wprowadziliśmy także zmiany w funkcjonalności naszego programu pozwalające na wygodniejsze korzystanie z gry. W trakcie testów wykryty został także drobny błąd, który szybko został usunięty.

Trzeciego dnia część zespołu zaczęła już pisać sprawozdanie, pozostała część udoskonalała wygląd gry. Ustaliliśmy, że przez dwa następne dni dalej będziemy zajmować się pisaniem sprawozdania.

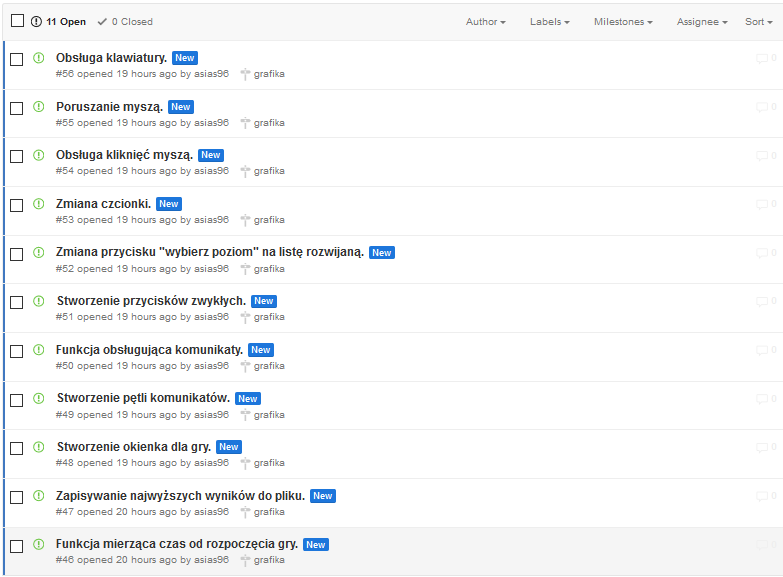
4. Plany na kolejne sprinty

Aktualna wersja gry, chociaż jest już funkcjonalna, może wciąż być rozwijana. Gdyby nasz projekt trwał dłużej, wprowadzalibyśmy do niej kolejne ulepszenia, pamiętając, by na koniec każdego z kolejnych sprintów dostarczyć gotowy do potencjalnego wydania produkt.

W kolejnych sprintach skupilibyśmy się przede wszystkim na dołączeniu do gry biblioteki graficznej (przejście z konsoli do programu okienkowego), co uczyniłoby środowisko gry bardziej przyjazne użytkownikowi, a także na dodaniu możliwości odkrywania lub zaznaczania pól przy pomocy myszki. Ponadto chcielibyśmy dołączyć pomiar czasu gry oraz możliwość eksportowania swoich wyników (oraz pseudonimu gracza) do pliku, zawierającego najlepsze czasy gry na poszczególnych poziomach.

Dodatkowo chcielibyśmy, aby pierwsze odkryte pole nie mogło być bombą (aby w pierwszym kroku nie dało się przegrać). Ciekawymi pomysłami wydają nam się także: wykorzystanie efektów dźwiękowych, np. w razie przegranej i możliwość *resetowania* planszy w dowolnym momencie gry.

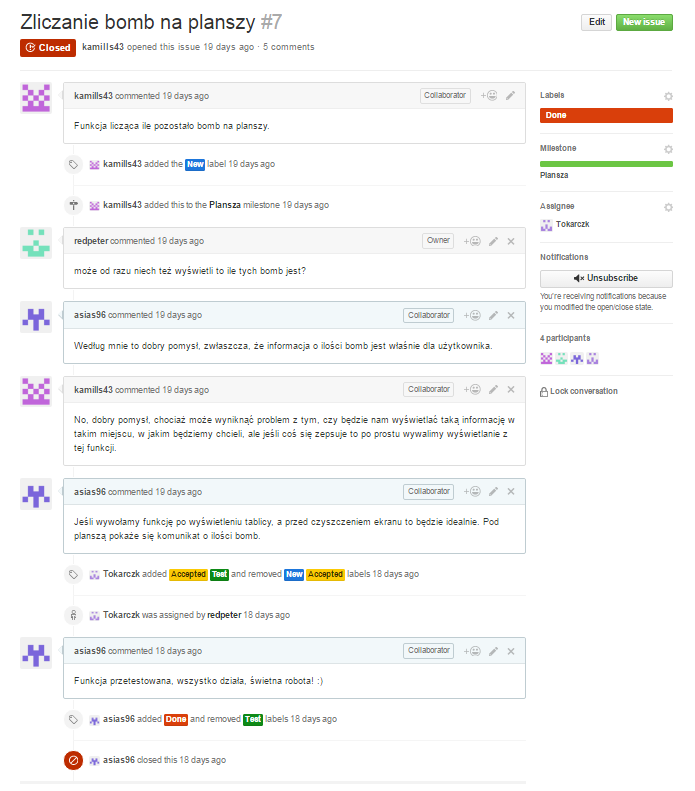
Mamy jednak na uwadze, że przed przystąpieniem do kolejnego sprintu musielibyśmy go zaplanować - wybrać jego główny cel i oszacować, ile jesteśmy w stanie zrobić w przyjętym czasie (biorąc pod uwagę dotychczasowe tempo prac), dlatego obecnie nie jesteśmy w stanie jednoznacznie wskazać, ile oraz które z powyższych udogodnień uwzględnilibyśmy w najbliższym sprincie. Możliwe, że w trakcie planowania sprintu okazałoby się, że któreś z tych zadań mają inny niż by się to mogło na pierwszy rzut oka wydawać poziom trudności, w związku z czym bylibyśmy zmuszeni do przełożenia części zadań na kolejne etapy rozwoju gry lub - w wersji optymistycznej - moglibyśmy zdecydować o podjęciu się realizacji większej ich liczby. Mając na uwadze prawdopodobne zawieszenie prac nad projektem w najbliższym okresie postanowiliśmy do backlogu produktu, jakim jest nasza gra dodać omawiane wyżej funkcjonalności:



Podsumowanie pracy zespołowej.

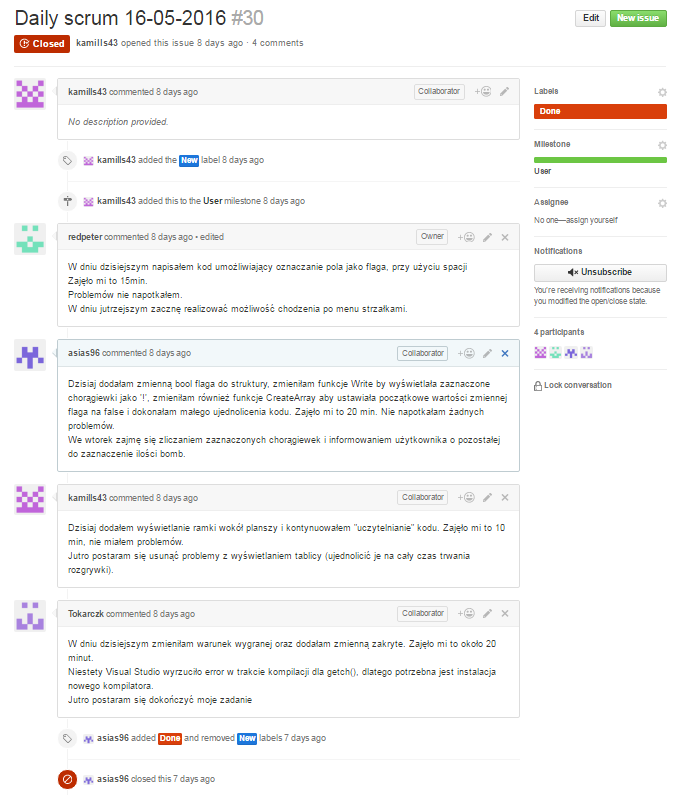
Już na początku wspólnej pracy ustaliśmy pewne zasady, które towarzyszyły nam przez pozostały czas. Każdy sprint był dokładnie przez nas zaplanowany. Mieliśmy jasno postawiony cel i wyznaczone do jego osiągnięcia zadania. Na koniec każdego sprintu staraliśmy się dostarczyć odbiorcy działający produkt. Współpraca przebiegała bardzo pomyślnie i bezkonfliktowo dlatego też, już po trzech sprintach możemy pochwalić się w pełni działającym programem, wzbogaconym o tyle, na ile wystarczyło nam czasu.

Codziennie każdy z nas pracował nad przydzielonym mu zadaniem sumiennie je wykonując. W razie problemów jednego z członków zespołu pozostali zawsze służyli pomocą. Nie zawodziła organizacja. Komunikacja między nami również przebiegała sprawnie, co jest kolejną przyczyną osiągniętego sukcesu. Do komunikacji wykorzystywaliśmy popularny serwis społecznościowy facebook.com jak i również github.com, umieszczając komentarze pod danymi issue i dyskutując poruszone kwestie w nich kwestie.

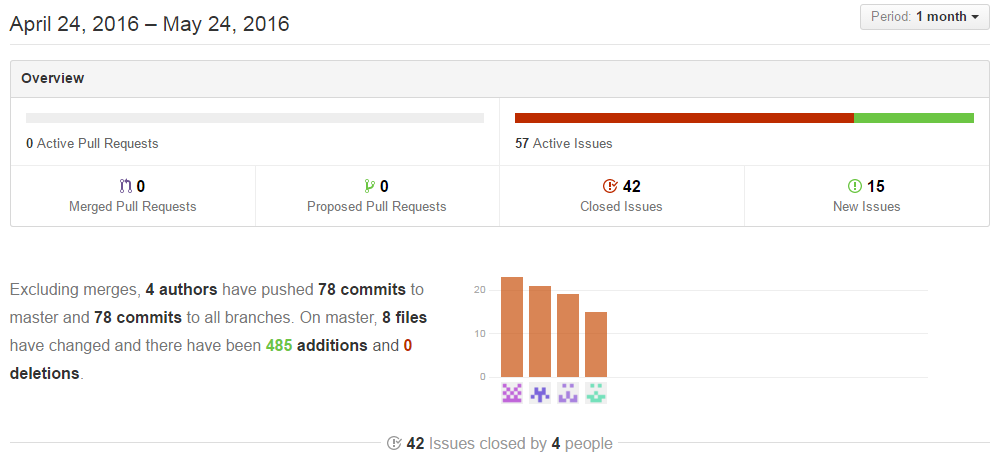


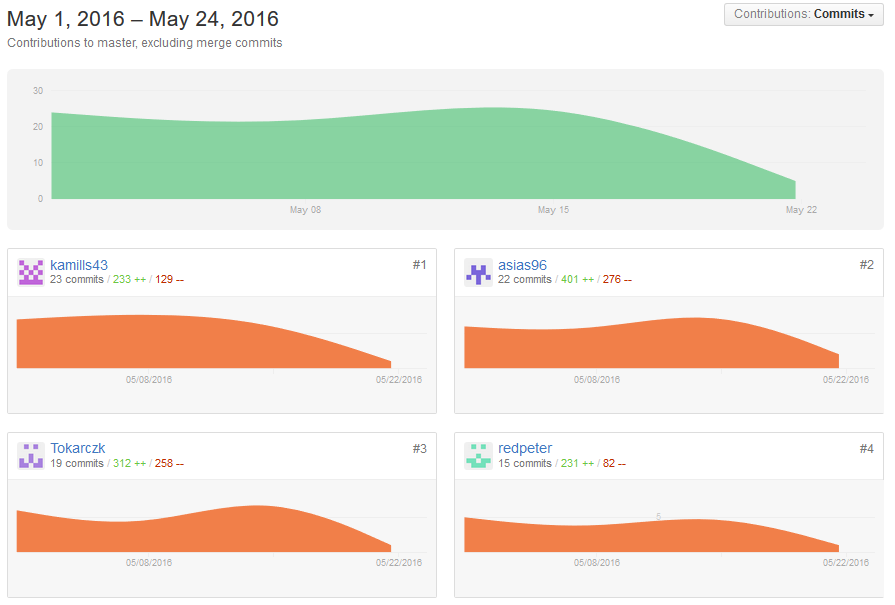
W naszym zespole osoba, która była Scrum Masterem była odpowiedzialna za czuwaniem nad przebiegiem prac i nad dotrzymywaniem terminów. Pilnowała ona także, aby w każdym sprincie osiągnąć cel, poprzez odpowiednie nakierowanie zespołu na działania związane z milestonem.

Każdego dnia pracy “spotykaliśmy się” na daily scrumach, tworząc odpowiedni issue z datą, przypisany do danego milestona, pod którym umieszczane były komentarze: kto co zrobił w danym dniu, ile czasu poświęcił, jakie miał ewentualne problemy i jakie zadanie wykona następnego dnia.



Codziennie każdy z nas wykonywał przynajmniej jeden commit do repozytorium. Napotkaliśmy przy tym niewielką ilość konfliktów, które udawało się zawsze sprawnie rozwiązać. Wynikały one głównie z sytuacji, gdy dwie osoby pracowały nad tym samym zadaniem w tym samym czasie. Rozwiązywaniem zajmowała się osoba u której konflikt powstawał.





Zakończenie:

Projekt ten pomógł nam poznać w praktyce metodykę Scrum. Nauczyliśmy się jak dzielić pracę w czasie oraz jaką jej część przyznawać sobie na każdy dzień. Zauważyliśmy, że podział na sprinty bardzo dobrze porządkuje cele, które są bardziej motywujące niż zadanie docelowe jakim jest stworzenie gry. Niejednokrotnie doceniliśmy także rolę scrum mastera, który jest bardzo pomocny zwłaszcza w chwilach gdy powstają jakieś problemy oraz gdy trzeba pilnować dotrzymania terminów. W systematyce pracy pomaga także przyznawanie każdej osobie ticketów, które w jasny sposób wskazują co danego dnia powinniśmy zrobić.

W związku z korzystaniem z serwisu Github.com, w celu przeprowadzania commitów, korzystaliśmy z systemu kontroli wersji Git oraz z nakładki TortoiseGit. Pozwoliło nam to lepiej zrozumieć jak działa oprogramowanie służące do śledzenia zmian w realizowanym projekcie. Poznaliśmy również sposoby na radzenie sobie z konfliktami powstałymi w związku z jednoczesnym edytowaniem kodu przez dwie osoby.