

Usunięto z pierwotnej wersji programu 32 z 45 *„code smells”.* Po wstępnym zapoznaniu się z ostrzeżeniami pokazywanymi przez gem *Reek*, postanowiono rozpocząć ich usuwanie od pliku *word.rb*. W tym pliku znajdowało się początkowo 8 ostrzeżeń:

*word.rb*



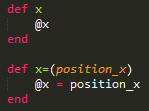
**Attribute:**

Ostrzeżenie to pojawia się w momencie, gdy użyto *attr\_accessor* na początku klasy. Używany jako getter oraz seter. Rozwiązaniem pozwalającym pozbyć się niechcianego „zapachu” jest rozbicie go na metody. Poniżej przedstawiono dokładny przebieg usuwania tego ostrzeżenia. Analogicznie postąpiono z pozostałymi atrybutami.

Przed:

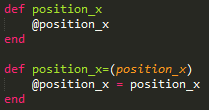
****

Po:



**UncommunicativeMethodName & UncommunicativeVariableName:**

W projekcie występowało wiele nic niemówiących dla czytającego kod programisty zmiennych i metod. Poniżej pokazano jeden przykład. Należało pamiętać by wprowadzić tą zmianę w całym kodzie.

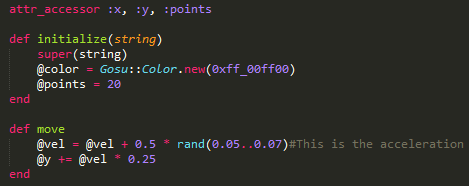
****

Z pliku *word.rb* pozbyto się 7 ostrzeżeń. Następną klasą, która została edytowana była klasa CompoundWord z pliku *compound\_word.rb*

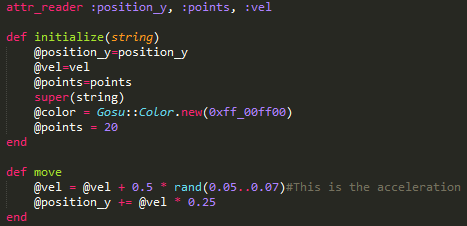
*compound\_word.rb*

Pozbyto się wszystkich ostrzeżeń występujących w klasie CompoundWord. Tym razem poza ostrzeżeniami Attribute oraz UncommunicativeVariableName wystąpiły takie ostrzeżenia jak: IrresponsibleModule oraz InstanceVariableAssumption. Pierwszy z nich mówi o tym, że klasy i moduły to jednostki, ponownego użycia, a więc warto dodać przynajmniej jeden krótki komentarz opisujący przeznaczenie klasy/modułu. Drugi z „nowych” ostrzeżeń informuje nas o tym, że zmienne instancyjne nie powinny być ustawiane poza definicją klasy, dlatego trzeba je zdefiniować w klasie. Cała klasa CompoundWord jest dość krótka, składa się zaledwie z 1 metody, więc całość przed edycja jak i po została zamieszczona poniżej.

Przed:

****

Po:

****

Kolejnym badanym plikiem był *document\_parser.rb.* Plik był stworzony przez grupę w celu dodania możliwości wyboru miedzy plikami o rozszerzeniu .txt oraz .docx. Początkowo występowały dwa ostrzeżenia z gemu Reek: InstanceVariableAssumption, IrresponsibleModule, jednak bazując na zdobytym doświadczeniu szybko się ich pozbyto. Przy okazji usuwania drugiego ostrzeżenia, usunięto go również z pliku *z\_order.rb*. Ostatnią badaną klasą była klasa GameWindow z pliku main.rb.

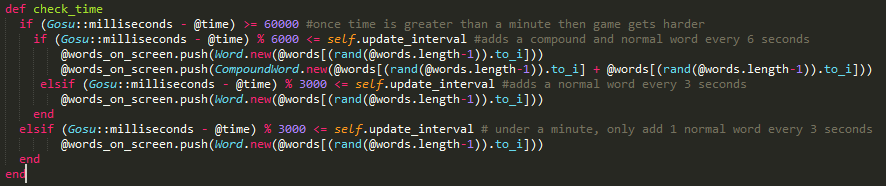
*main.rb*

W *main.rb* głównymi ostrzeżeniami, z którymi należało się uporać było ostrzeżenia typu: TooManyStatements oraz DuplicateMethodCall, również wystąpiły podobne ostrzeżenia jak w poprzednich klasach.

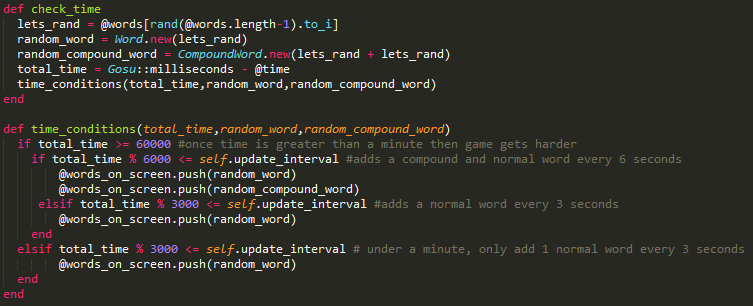
**TooManyStatements &** **DuplicateMethodCall:**

**„**TooManyStatements” jest to ostrzeżenie dotyczące długości metody. W dokumentacji *reek* można zauważyć, iż metoda powinna składać się z maksymalnie pięciu instrukcji. Nie zawsze jest to możliwe, szczególnie przy refaktoryzacji czyjegoś kodu. Często metody są długie, zbyt długie. Należy je rozbić na mniejsze metody. Dzięki temu kod na pewno będzie łatwiejszy do czytania oraz edycji dla innych programistów. Podczas refaktoryzowania projektu zdarzyło się nie pozbyć tego „zapachu”, a jedynie zredukować maksymalną liczbę instrukcji w poszczególnych metodach do sześciu. Niestety nie zawsze jest to możliwe bez większego naruszenia struktury kodu. Poniżej przedstawiono metody z pliku *main.rb,* z których usunięto ostrzeżenie „TooManyStatements”.

Przed:

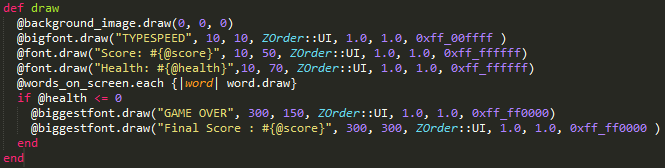
****

Po:

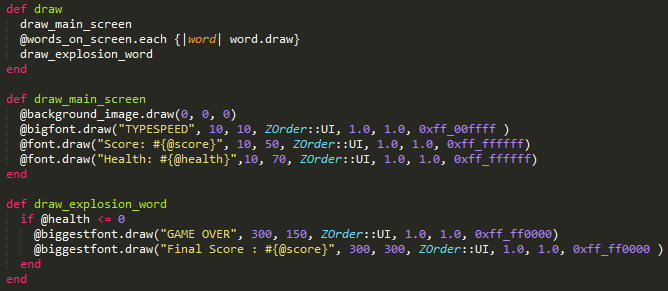
****

Powyższa metoda *check\_time* zawierała zbyt dużo instrukcji, a także zbyt wiele powtórzeń. Stworzono dodatkową metodę, a także zmienne, dzięki temu kod jest bardziej czytelny.

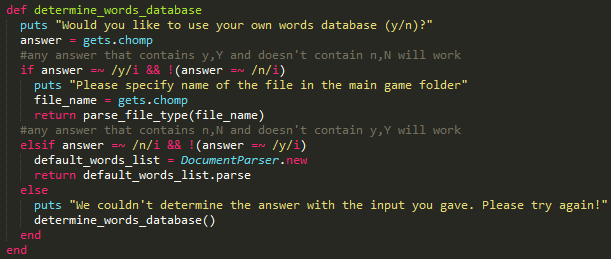
Przed:

****

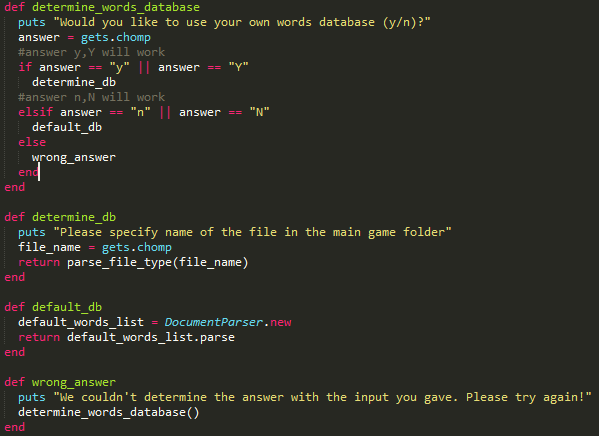
Po:

****

Przed:

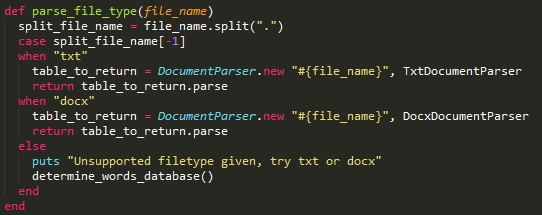
****

Po:

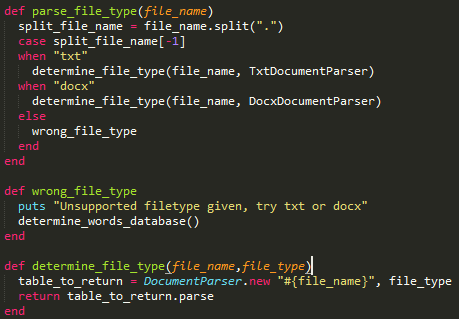
****

Metoda *determine\_words\_database* została rozbita, na mniejsze metody, które odpowiadają za: ustalanie nowego pliku, z którego mają być pobrane słowa do gry, wybór podstawowej bazy danych słów oraz za ponowne wykonanie metody w przypadku błędnej odpowiedzi użytkownika. Ponadto zmieniono warunek w instrukcji warunkowej, tak aby tylko „y lub Y” bądź „n lub N” były uznawane za poprawną odpowiedź. W pierwotnej wersji odpowiedź np. „yyyyy” spełniała pierwszy warunek.

Przed:

****

Po:

****

Metoda *parse\_file\_type* została rozbita na dwie dodatkowe, z których jedna odpowiada za ustalenie typu pliku jaki ma być parsowany.