
Raport

(środowisko agenta i reprezentacja wiedzy)

Marcin Woźniak
Filip Izydorczyk
Hubert Wrzesiński
Przemysław Fierek

Grupa 5

15 maja 2019

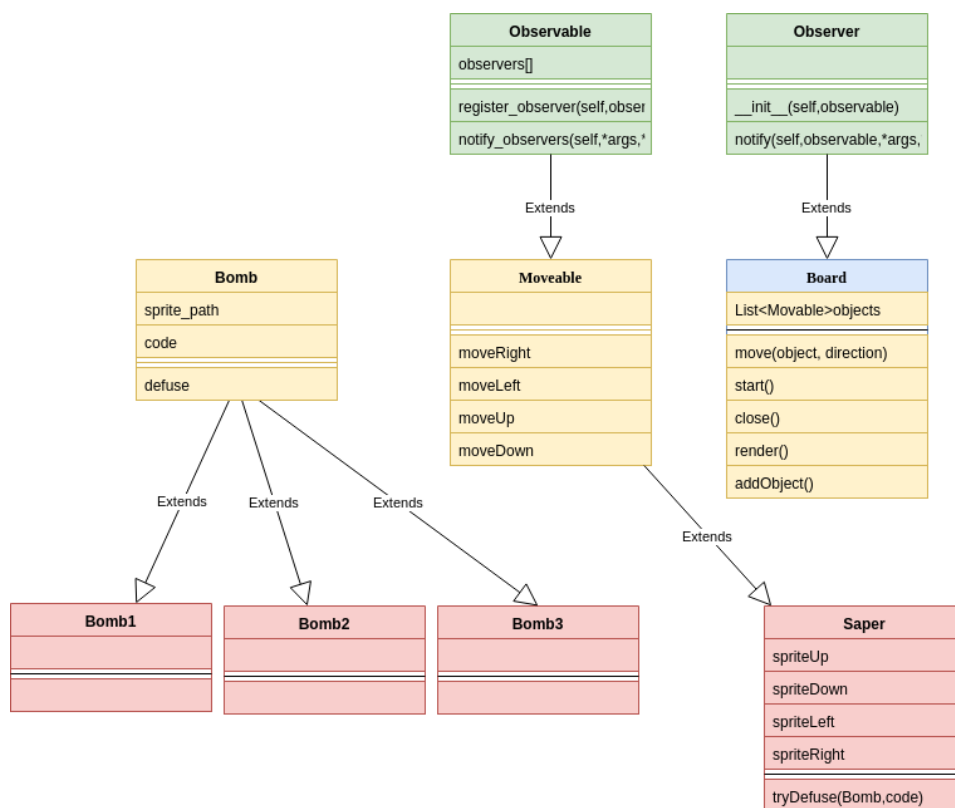
Repozytorium: https://github.com/linux923344/autonomiczny_saper/

Spis klas użytych w projekcie:

1. **Board** - jako argumenty przyjmuje X i Y, które są rozmiarem renderowanego okna. Klasa odpowiada za renderowanie mapy i zarządzanie obiektami na niej.
2. **Bomb** (BombRed, BombBlue, BombYellow) - obiekt odpowiedzialny za renderowanie oraz logikę bomby
3. **Direction** - enum, który daje nam informacje, w którym kierunku ma się poruszać postać.
4. **MapReader** - argumentem przy jej tworzeniu jest plansza czyli obiekt typu Board, na której MapReader będzie tworzył obiekty które są przechowywane w zewnętrznym pliku.
5. **Stone** - klasa, która jest odpowiedzialna za informacje potrzebne do wyrenderowania kamienia.
6. **Tool** - klasa, która jest odpowiedzialna za informacje potrzebne do wyrenderowania narzędzia.
7. **Saper** - przechowuje informacje potrzebne do wyrenderowania Sapera, co posiada w ekwipunku oraz jakie zadania musi wykonać.
8. **PathFinder** - tworzy klasę typu Graph na podstawie podanego Boarda
9. **Graph** - klasa znajduje ścieżkę w grafie do danego punktu i zwraca listę kierunków, w jakich musi poruszać się postać aby dojść do danego punktu.

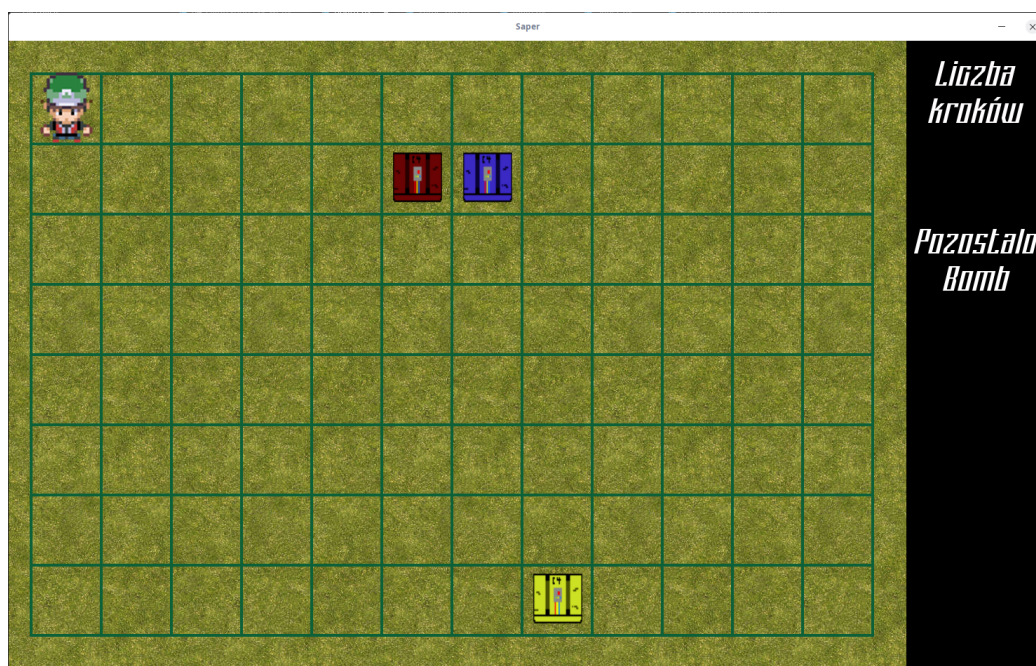
27.03.2019 r.

1. Zaprojektowanie diagramu klas.



2. Na podstawie diagramu, została stworzona klasa główna oraz jej podklasy.

3. Został stworzony szablon planszy razem z umieszczonymi na niej bombami.



Na załączonym zrzucie ekranu widoczny jest saper czyli nasz Agent, oraz trzy kolorowe obiekty (bomby), które nasz agent ma za zadanie je rozbroić.

10.04.2019 r.

W tym dniu została dokończona reprezentacja wiedzy w naszym projekcie. Przedstawimy ją teraz:

1. Posiadamy trzy rodzaje bomb (czerwona, żółta, niebieska).
2. Aby rozbroić bombę czerwoną musimy posiadać narzędzie dodatkowo zmieścić się w wyznaczonym czasie. Jednostka czasu w naszym świecie jest jeden krok.
3. Aby rozbroić pozostałe bomby Saper musi tylko do nich podejść, oczywiście jak najkrótszym czasie.
4. Saper będzie wyznaczał drogę za pomocą algorytmu przeszukiwania grafu.

15.04.2019 r.

W tym dniu został zaimplementowany wyznaczanie ścieżki za pomocą algorytmu przeszukiwania DFS oraz kilka klas.

17.04.2019 r.

Spisanie raportu i wypisanie wszystkich klas znajdujących się w projekcie.

30.04.2019 r.

Dodanie ekwipunku i ustalenie zasad działania bomb. Bomby będą miały określony czas, a narzędzia będą potrzebne do rozbrowienia ich.

6.05.2019 r.

Implementacja algorytmu BFS.

14.05.2019 r.

Implementacja algorytmu Best-First Search.