

# PRPG-combat

---

## Pojedynek prowadzony na zasadach gier typu PRPG

Program symuluje walkę którą można odbyć podczas gry w D&D, lub Warhammera. W takich rozgrywkach obok decyzji gracza, przebieg zdarzeń determinuje rzut kością. Tutaj będzie to generator liczb losowych. Zarówno gracz jak i jego przeciwnik posiadają grupę statystyk takich jak inteligencja, siła etc. Zależą one jednocześnie od decyzji gracza co do klasy postaci, jak i od rzutu kością. Gra kończy się gdy hp jednej z postaci spadnie do 0.

Klasa **stats** ma za zadanie przechowywanie podstawowych statystyk postaci. Jej konstruktor przyjmuje zmienną typu int. Stanowi ona wybór co do klasy postaci dla której statystyki mają być wylosowane. Zawiera w sobie metody odpowiedzialne za zwracanie statystyk.

Losowania opierają się na funkcji **diceroll**, która przyjmuje dwie liczby typu int. Pierwsza określa zasięg losowania (od 1 do podanej liczby). Druga określa ile „rzutów kością” ma zostać wykonanych. Funkcja zwraca sumę otrzymanych losowań.

Klasa **creature** jest klasą bazową dla klas NPC oraz Player. Zawiera grupę zmiennych takich jak HP (zdrowie), MP (maksymalna ilość ruchów w rundzie). Jej konstruktor również przyjmuje liczbę typu int, która jest odpowiednikiem profesji postaci analogicznie do klasy stats.

Metoda **look** w klasie creature przyjmuje inny obiekt typu creature i zwraca odległość między nimi wyrażoną w ruchach na planszy.

Klasa **Player** dziedziczy po klasie creature i dzieli z nią konstruktor. Zawiera w sobie unikalną metodę **move**, która zmienia współrzędne x i y. Przyjmuje ona cyfrę typu int i zmienia położenie postaci analogicznie do położenia klawiszy na klawiaturze numerycznej. Np. 7 wykonałoby ruch po skosie w lewo i do góry.

Metoda **statusbar** w klasie Player drukuje na ekranie dane potrzebne do rozgrywki HP, MP, range, współrzędne i odległość od przeciwnika.

Metoda **fleecheck** w klasie Player sprawdza, czy Player nie opuścił wyznaczonego terenu. Jeśli to zrobił zwraca zmienną logiczną o wartości true. W przeciwnym przypadku false.

Klasa **NPC** dziedziczy po klasie creature i dzieli z nią konstruktor. Zawiera w sobie metodę **seek**, która przyjmuje zmienną typu creature. Metoda ta ma za zadanie

sprawdzić, czy podany obiekt jest w zasięgu NPC i w przeciwnym przypadku przesunąć go po najkrótszej drodze tak blisko podanego obiektu na ile pozwalają jego MP.

Klasy NPC i Player dzielą ze sobą następujące metody:

Metoda **attack** przyjmuje zmienną typu creature. Jeśli podany obiekt jest w zasięgu, wykonuje próbę ataku, która opiera się na funkcji diceroll i statystykach atakującego. Przy udanym ataku zmniejsza HP podanego obiektu i drukuje komunikat o zadanych obrażeniach. W przypadku nieudanego ataku, lub bycia poza zasięgiem ataku, wyświetla komunikat o nieudanej próbie jeśli została użyta w klasie Player, lub nie robi nic w przypadku klasy NPC.

Metoda **deathcheck** sprawdza czy HP jest równe, bądź mniejsze od 0. Jeśli jest to prawda, zwraca zmienną logiczną o wartości true i drukuje na ekranie komunikat o porażce (jeśli użyta w klasie Player), lub zwycięstwie (jeśli użyta w klasie NPC). W przeciwnym przypadku false.

Metoda **checkterrain** sprawdza współrzędne obiektu i określa czy porusza się w trudnym otoczeniu. Jeśli to prawda, zwiększa jego zużycie MP z każdym ruchem o 1, zmniejszając ilość ruchów które obiekt może wykonać w trakcie rundy. Jeśli użyta w klasie Player, drukuje dodatkowy komunikat.

Funkcja **profchoice** drukuje na ekranie komunikat proszący użytkownika o wybranie profesji ( liczby całkowitej z przedziału 1-3) i zwraca podaną liczbę.

Funkcja **game** przyjmuje zmienne typu Player i NPC, które będą uczestniczyły w pojedynku. Funkcja zakończy działanie gdy HP jednej z postaci będzie mniejsze, bądź równe 0, lub kiedy użytkownik opuści wyznaczone terytorium.

Funkcja **show** przyjmuje zmienne typu Player i NPC. Wyświetla na ekranie planszę po której będą się poruszać obiekty i ich pozycję na planszy.