

Dokument projektowy

Adrian Kula
Sylwester Naklicki
Krzysztof Rozmej
Łukasz Lebioda
Artur Kotyra

Link do JIRY: <https://projektzsp.atlassian.net/jira/software/projects/PROJ/boards/1/roadmap>

Motywacje

Lubimy gry komputerowe w stylu retro i dlatego zdecydowaliśmy się na stworzenie takiej gry. Praca nad stworzeniem gry komputerowej wydawała nam się interesująca, dlatego postanowiliśmy połączyć przyjemne z pożytecznym.

Zdecydowaliśmy się na Unity, ponieważ większość zespołu zna tą technologię w większym lub mniejszym stopniu. Ponadto, Unity jest przystępnym narzędziem i wykorzystuje język C# który nie jest zbyt skomplikowany. Dzięki temu będziemy w stanie sprawnie poradzić sobie z zadaniami.

Nasz projekt ma za zadanie zapewnić rozrywkę

Przedział czasowy

Pożądany: Pierwszy semestr (Projektowanie) : 01.10.2021 - 28.01.2022

drugi semestr (implementacja) : 28.02.2022 - 26.06.2022

Ewentualny: Pierwszy i drugi semestr + wakacje

Pożądane rezultaty i cel projektu

Rozwinięcie umiejętności pracy zespołowej

Rozwinięcie umiejętności zarządzania zasobami zespołu

Zdanie semestru

Nauka nowych technologii oraz poszerzenie wiedzy w technologiach nam znanych

Działająca gra z zaplanowaną wartością

Narzędzia i technologie

Unity - Silnik do gry

C# - Główny język kodowania

Json - System składowania danych

GitHub - Przechowywanie kodu

GoogleDrive - Przechowywanie plików wypuszczonych wersji gry (Release candidate)

Visual Studio 2022 - IDE

JIRA - Narzędzie do zarządzania projektem

Draw.io i StarUML - Narzędzie do tworzenia dokumentacji

Aseprite i Gimp - Narzędzia do tworzenia i edycji grafiki

Bosca ceoil, Audacity i LMMS - Narzędzia do tworzenia i edycji dźwięku

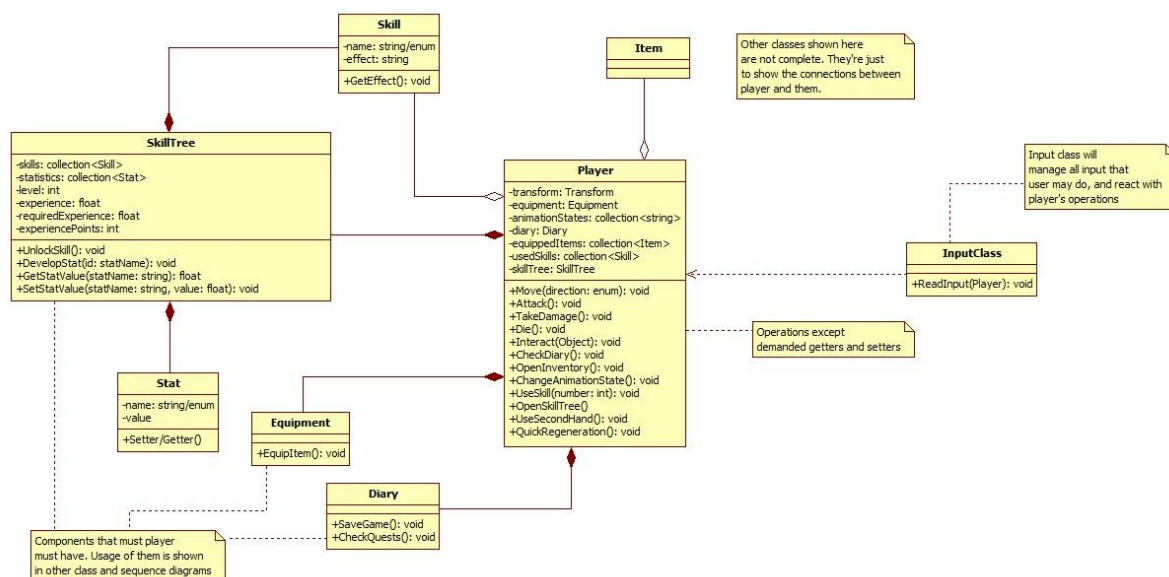
Discord- komunikacja :)

CI/CD

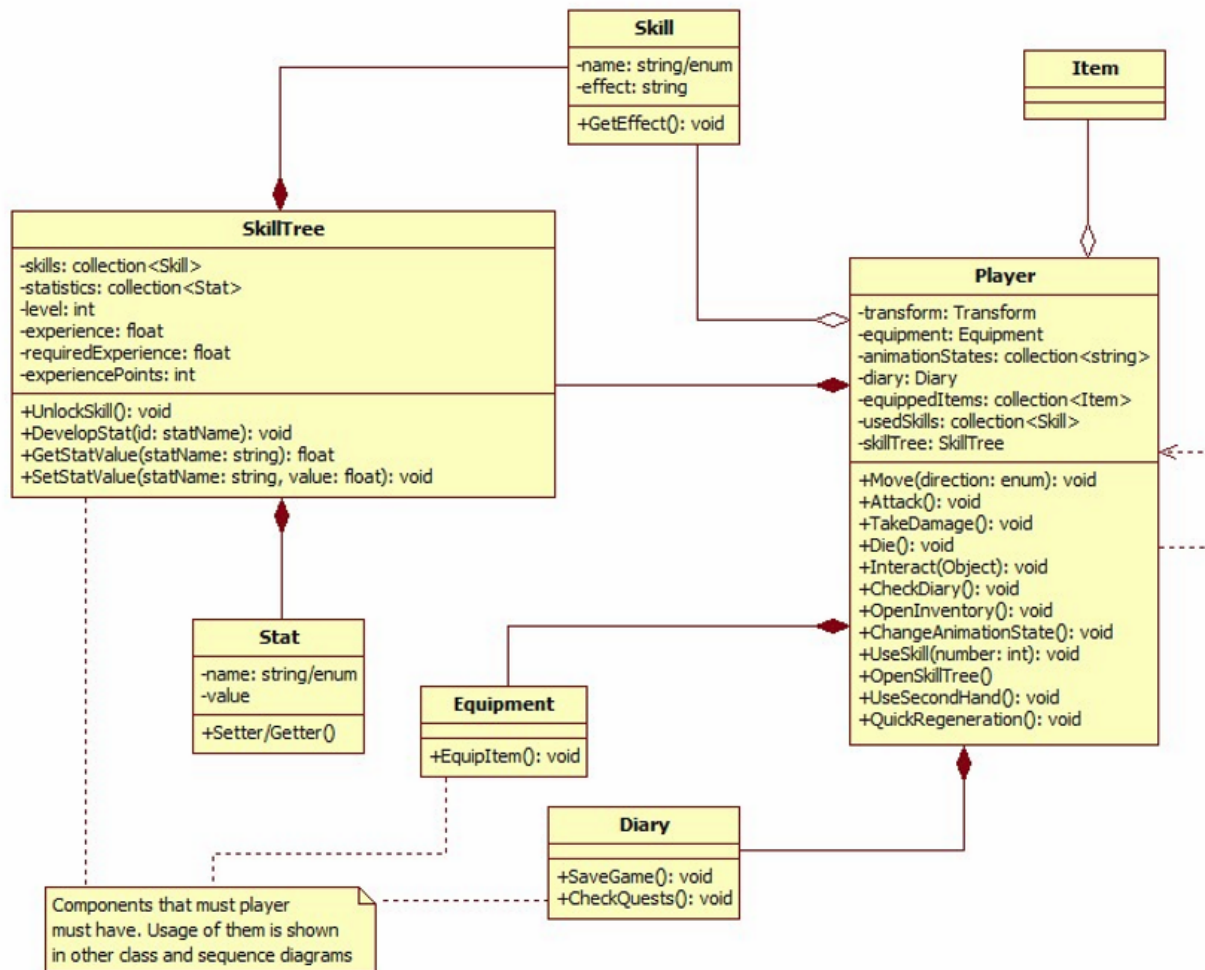
1. Wybranie tasków backlogu zgodnie z możliwościami zespołu
2. Dystrybucja tasków pomiędzy członkami zespołu
3. Praca i nad taskami
4. Testowanie tasków
5. Wypuszczenie wersji gry
6. Backlog refinement

Dokumentacja projektowa

1. Klasa Player



Klasa Player jest jedną z najistotniejszych klas, bowiem w niej zawierać się będą wszystkie atrybuty i operacje odnoszące się do postaci gracza i do interakcji gracza ze światem gry. Klasa ta zawiera w sobie oczywiście atrybut typu Transform będący obiektem wygenerowanym przez silnik i umiejscowionym w świecie gry. W klasę Player wkomponowane są obiekty typu Equipment, Diary, SkillTree, które odpowiadają kolejno za ekwipunek jaki gracz będzie posiadał, dziennik, służący do zapisu stanu gry i sprawdzania zadań oraz drzewko umiejętności odpowiadające za rozwój statystyk gracza. Wszystkie operacje jakie klasa Player posiada, mają za cel manipulację wartościami tych atrybutów przy użyciu wyżej wymienionych obiektów klas.



Natomiast klasa InputClass ma za zadanie przechwytywać input i wywoływać odpowiednie operacje gracza.

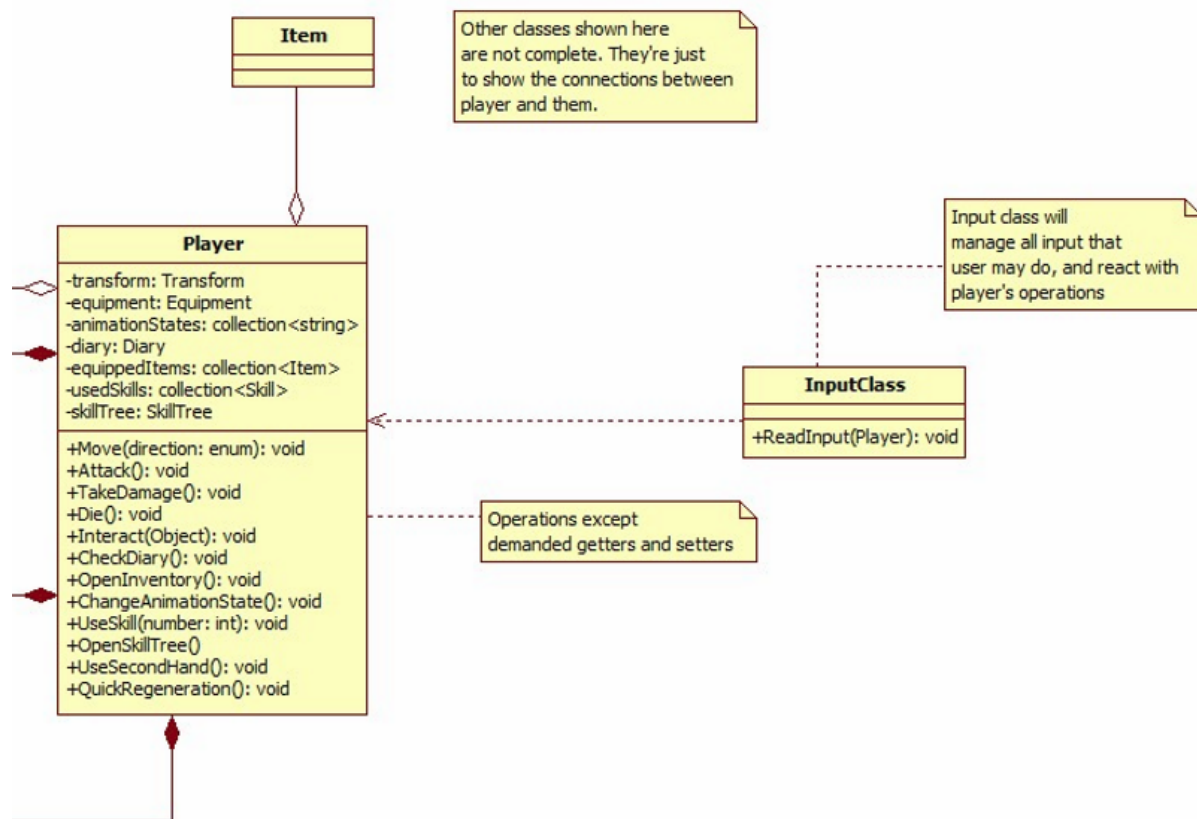
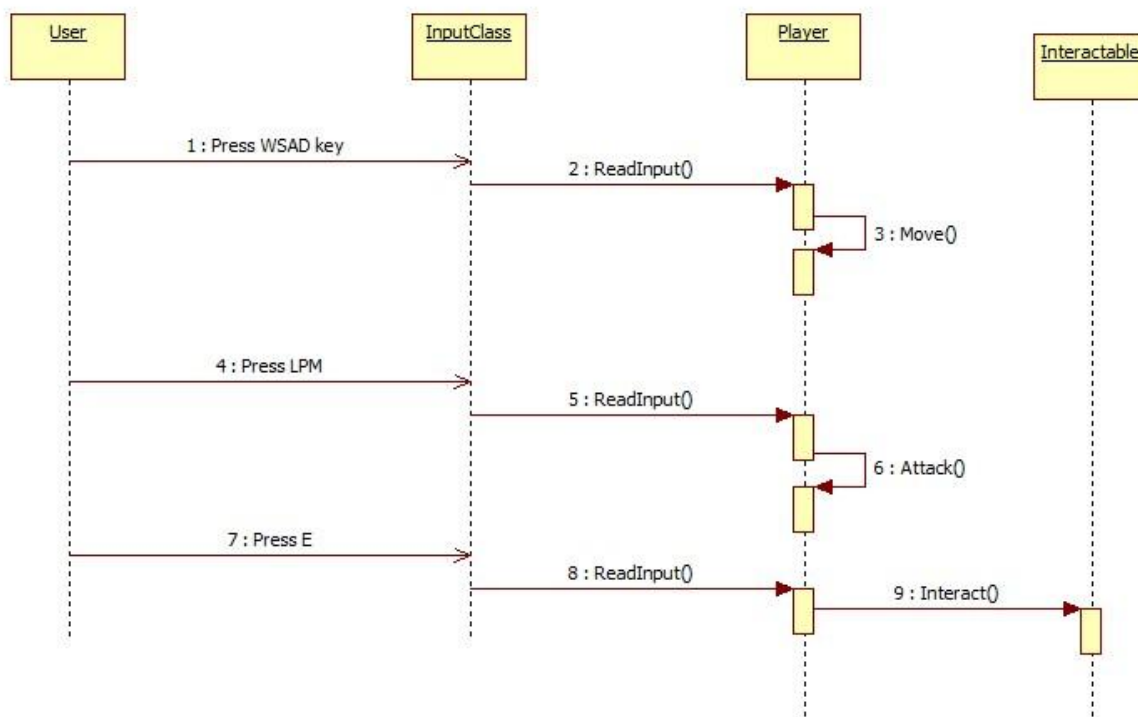
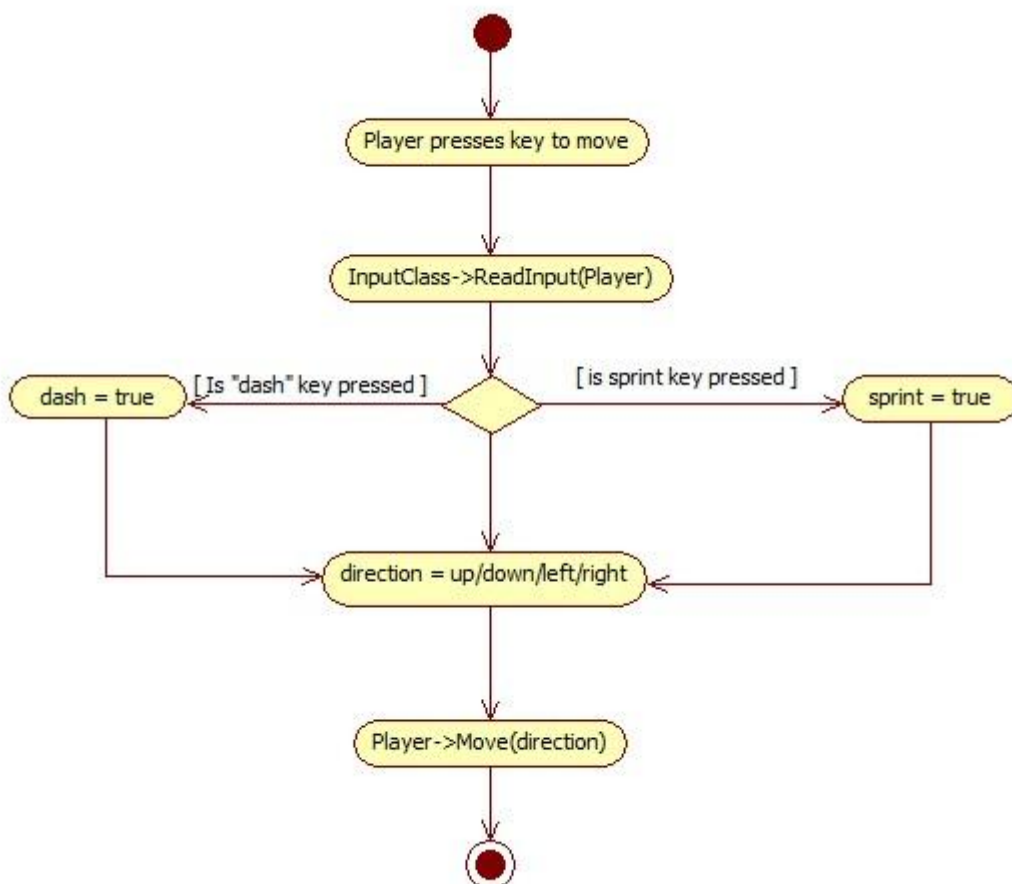


Diagram sekwencji obrazujący relację między InputClass oraz Player:

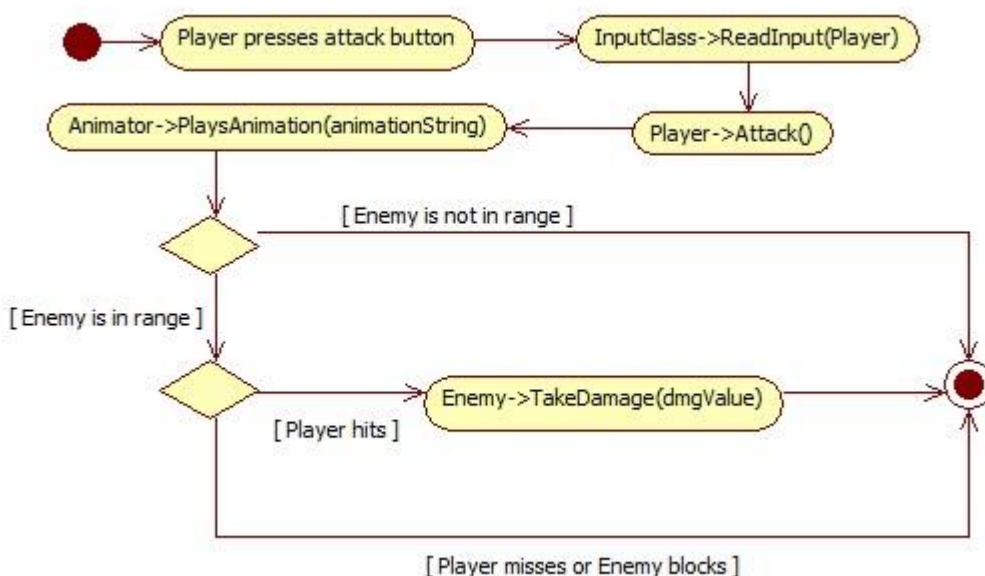


Jak łatwo zauważyć, gracz/użytkownik wciska odpowiedni klawisz na klawiaturze bądź przycisk myszki. Operacja ReadInput() przechwytuje kod naciśniętego klawisza i stymuluje wywołanie odpowiedniej, co do wciśniętego klawisza, operacji w klasie Player.

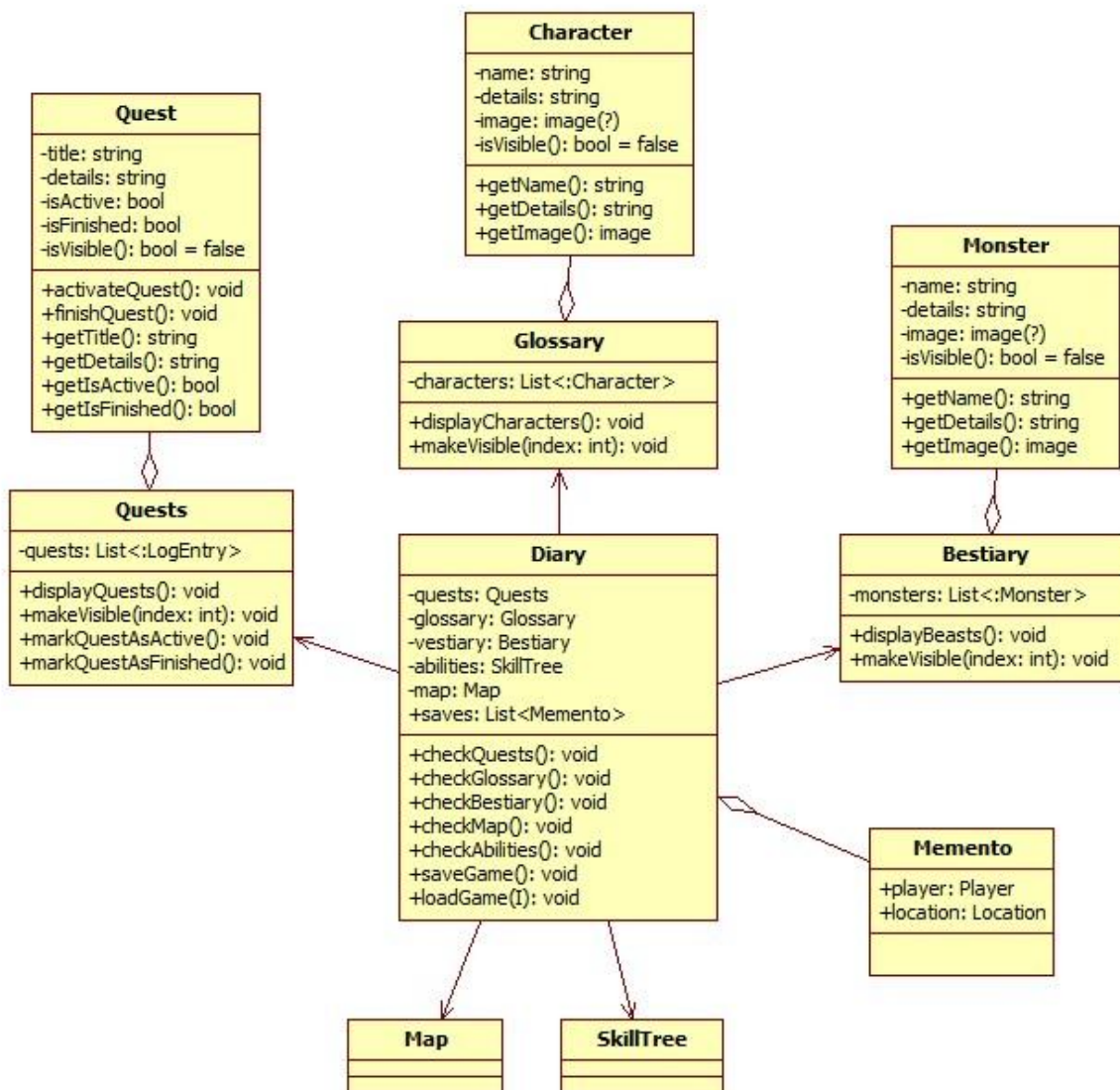
Poniżej mamy przedstawiony diagram aktywności przedstawiający proces prowadzący do poruszenia się gracza:



Podobnie przebiega proces odpowiadający za walkę z przeciwnikiem, pokazany na diagramie poniżej:



2. Klasa Diary



Klasa Diary zawiera w sobie dziennik zadań, jakie ma do zrobienia gracz.

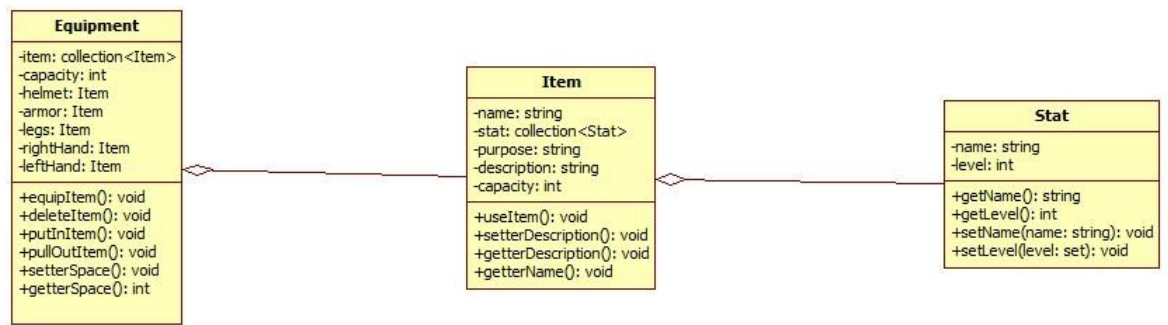
Zadania będą opisane wcześniej przez nas.

Glosariusz to opis postaci w świecie. Składa się on z klas typu Character a każda z nich ma własny opis, wygląd imię itd.

Bestiariusz jak wyżej, ale z potworami.

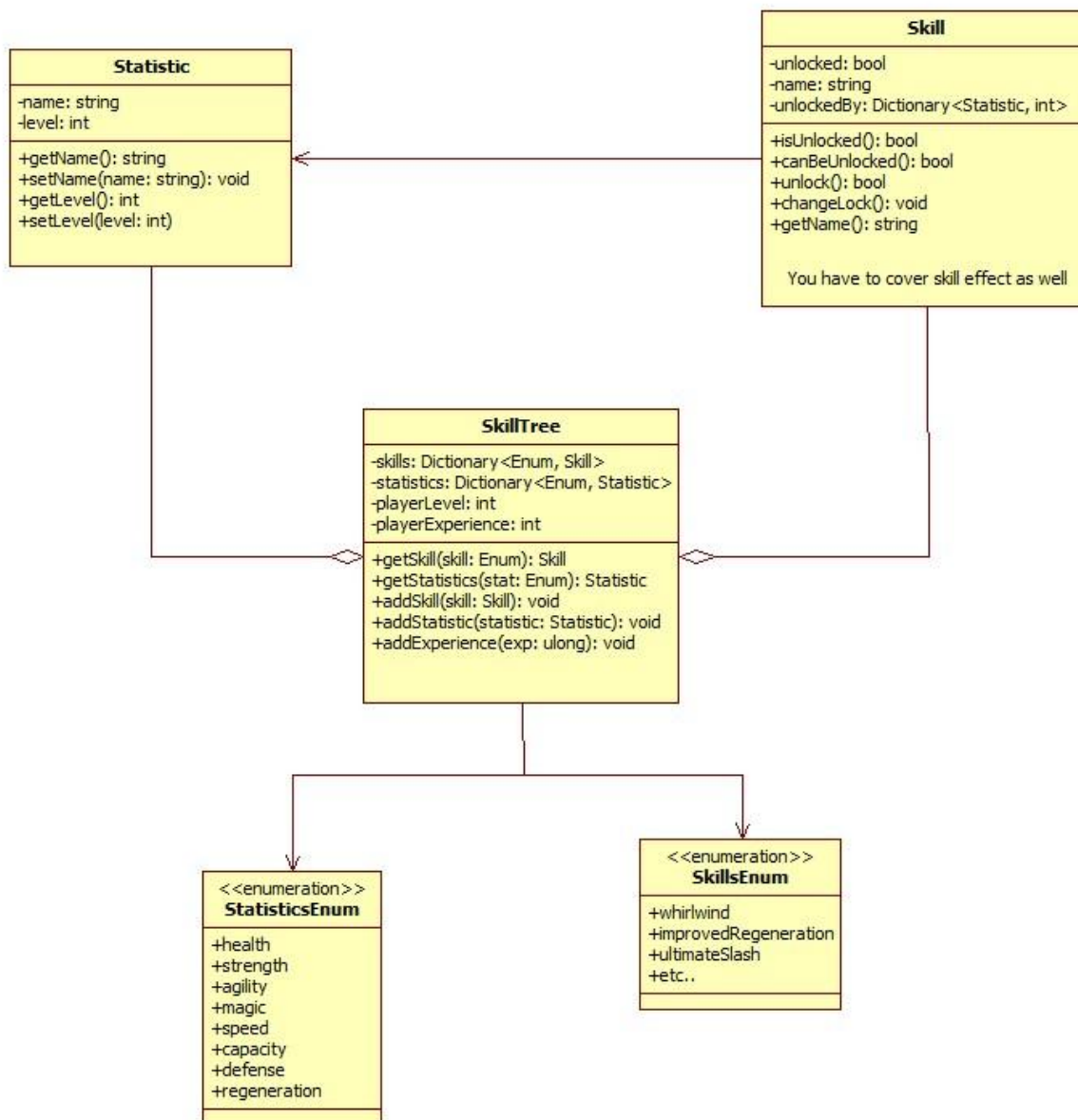
Ponadto, w dzienniku można sprawdzić mapę oraz zapisać czy wczytać stan gry

3. Klasa Equipment



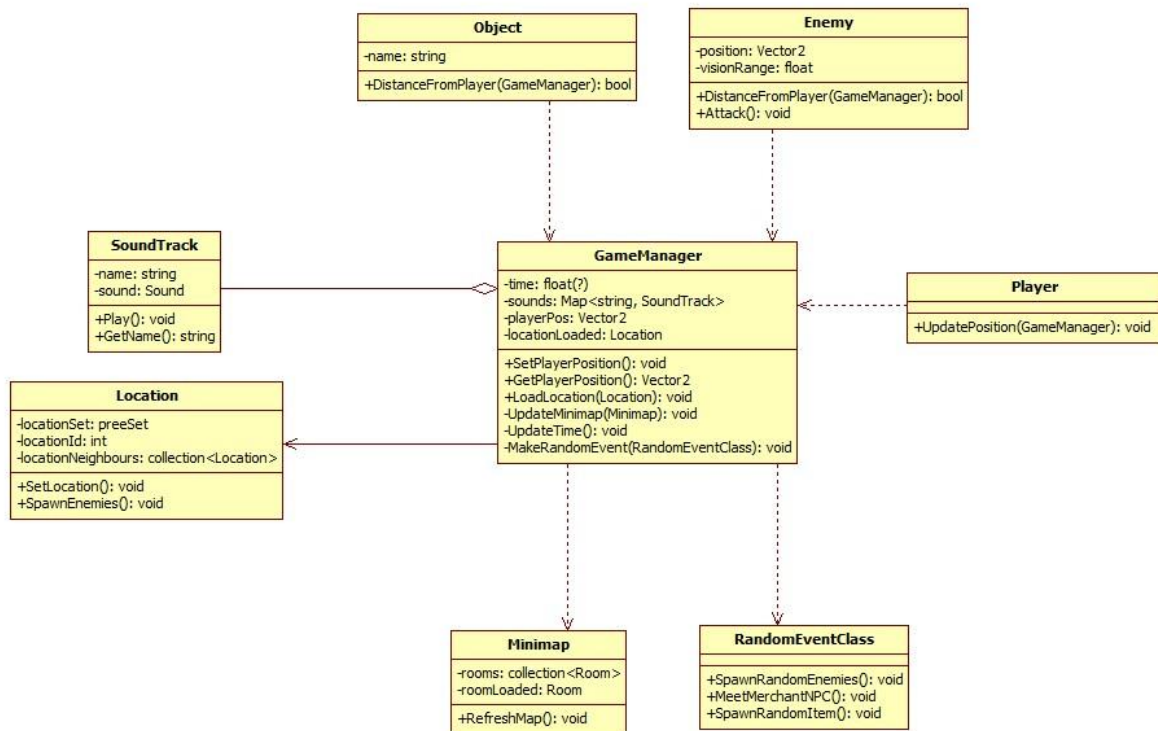
Klasa Equipment opisuje ekwipunek zarówno plecak i jak wyposażenie gracza. W ekwipunku gracz będzie mógł zakładać na siebie przedmioty jak i usuwać je z ekwipunku.

4. Klasa SkillTree

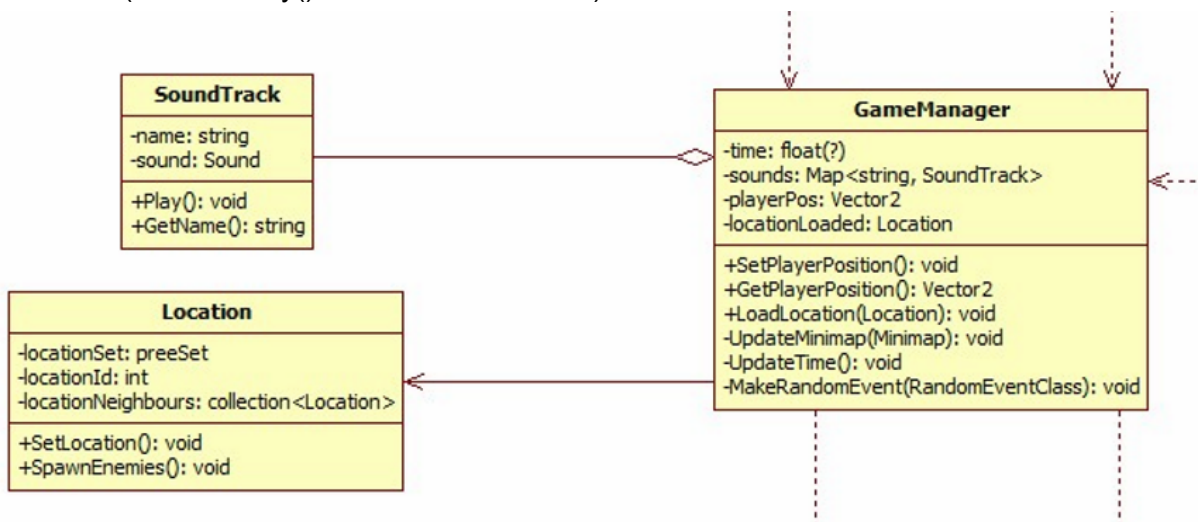


Klasa SkillTree odpowiada za zarządzanie statystykami oraz umiejętnościami postaci. Typy statystyk jak i umiejętności przechowywane będą w typach enumerowanych. Statystkami będą wartości natomiast umiejętnościami będą działania jakie może podjąć gracz. Odblokowywanie umiejętności odbywa się za pomocą zwiększania poziomu doświadczenia danej statystyki. Odblokowywanie statystyki odbywa się za pomocą wyboru jej po wcześniejszym zwiększeniu poziomu doświadczenia.

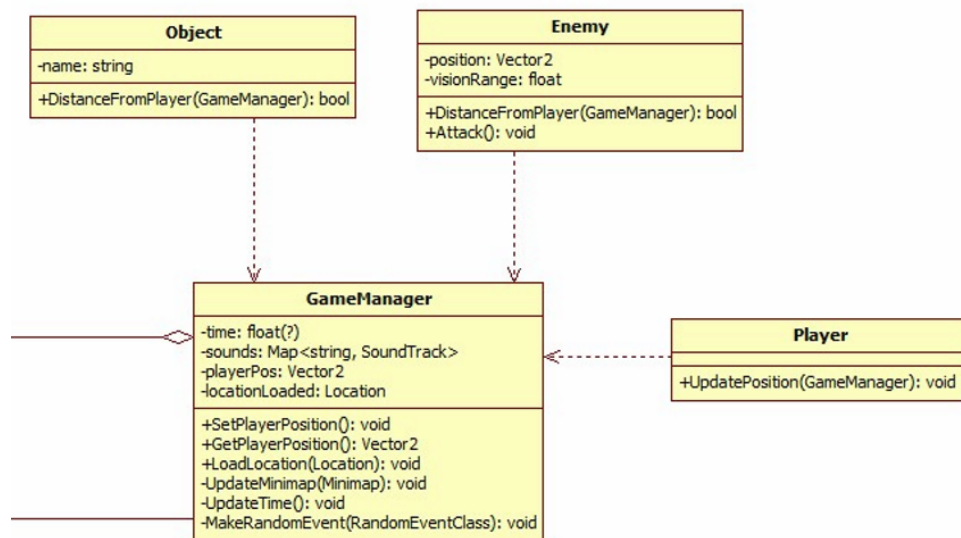
5. Klasa GameManager



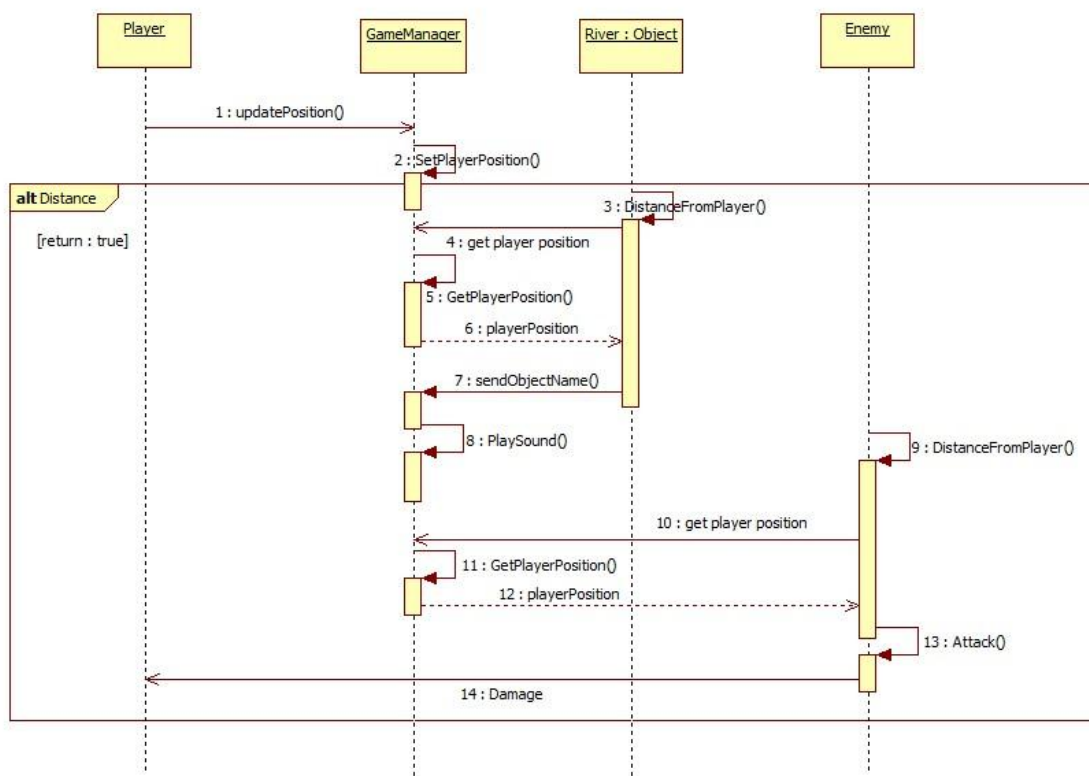
Przedstawiona powyżej klasa odpowiada za zarządzanie światem gry i jego zachowaniami. Ma ona za zadanie: ładować konkretne lokacje (LoadLocation()), odtwarzać dźwięki otoczenia (metoda Play() w klasie SoundTrack):



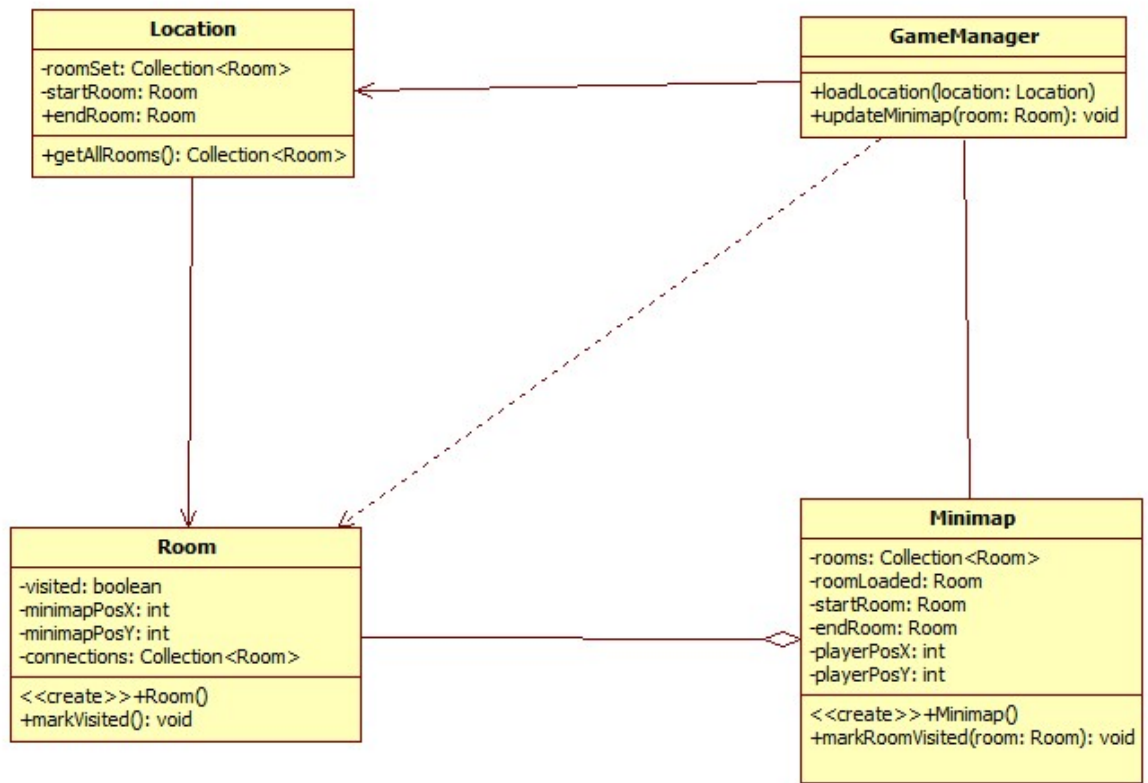
Aktualizować pozycję gracza w lokacji i udostępniać ją innym obiektom klas, takim jak przeciwnik (Enemy), obiekt znajdujący się w lokacji (Object) oraz minimapa (Minimap).



Implementuje ona wzorzec *Obserwator*, gdzie obserwatorem jest obiekt GameManager, a obserwowanym obiekt Player. Gdy gracz zmieni swoje położenie, wysyła je do GameManager'a, a ten aktualizuje jego pozycję oraz odświeża w ten sposób obraz wyświetlany na minimapie. W zależności od pozycji gracza względem innych obiektów, mogą zaistnieć różne zdarzenia. Jednym z nich może być zauważenie gracza przez przeciwnika, jeśli ich odległość od siebie będzie mniejsza niż pole widzenia przeciwnika. Poniższy diagram sekwencji obrazuje tę komunikację między GameManagerem, a innymi obiektami:

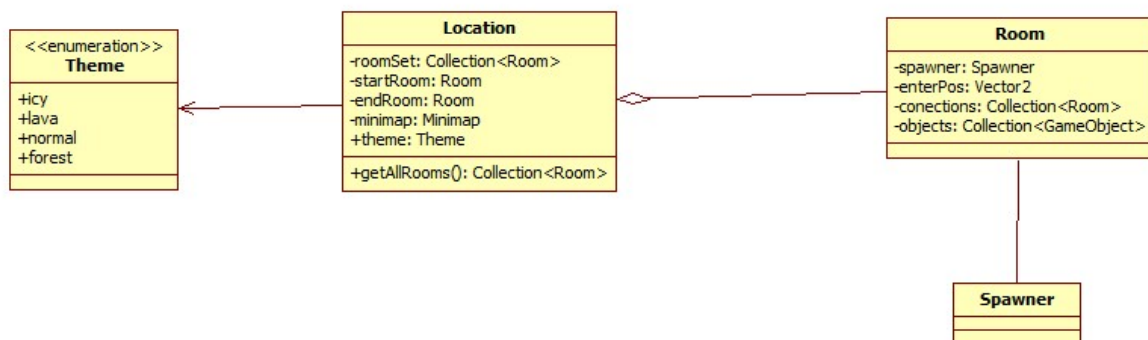


6. Klasa Minimap



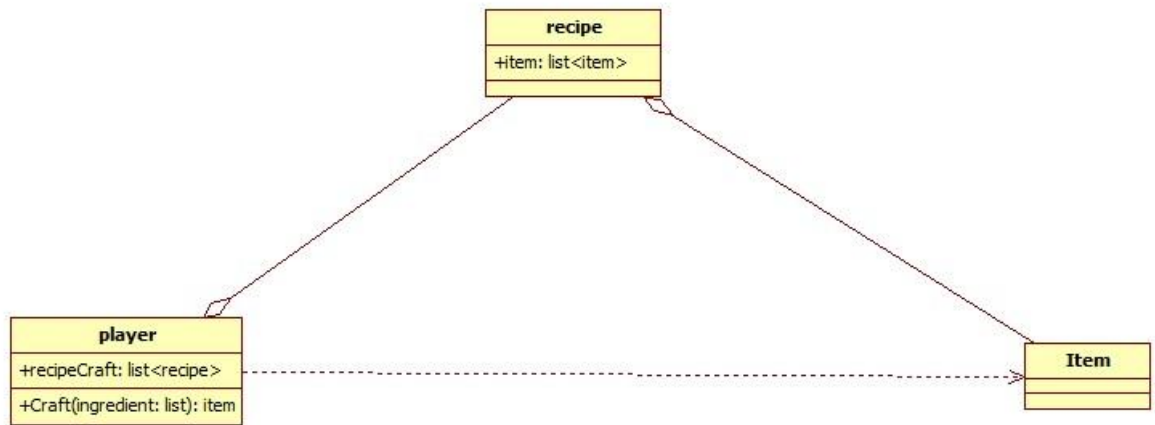
Klasa Minimap służy do zarządzania sposobem wyświetlania układu mapy w duncieonie lub w świecie. Układ minimapy w duncieonie będzie układem czterostronnym(tj. z każdego pokoju są maksymalnie 4 przejścia do następnego).

7. Klasa Location(duncieon)



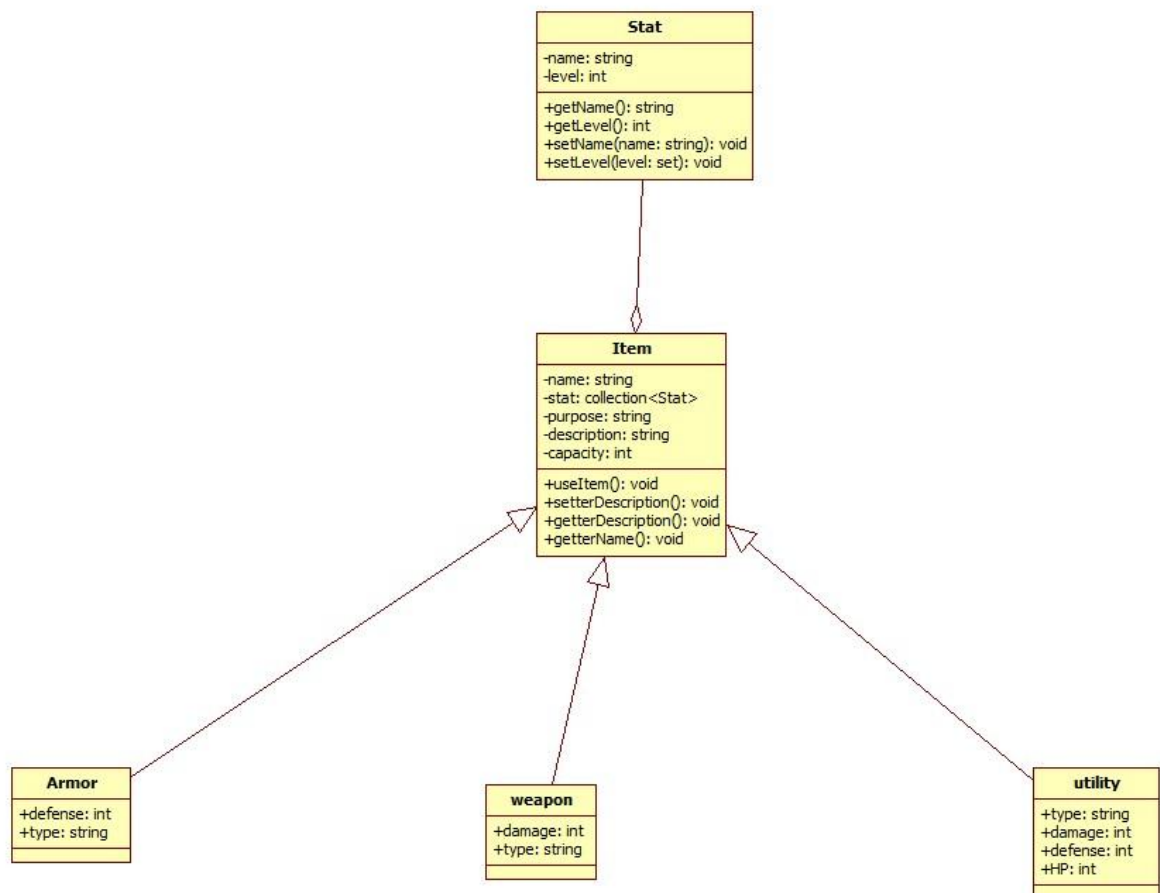
Powyższa klasa umożliwia tworzenie duncieonów z pokojami. Pokoje zawierają spawnery przeciwników oraz obiekty "martwe" otoczenia.

8. Klasa BuilderItem



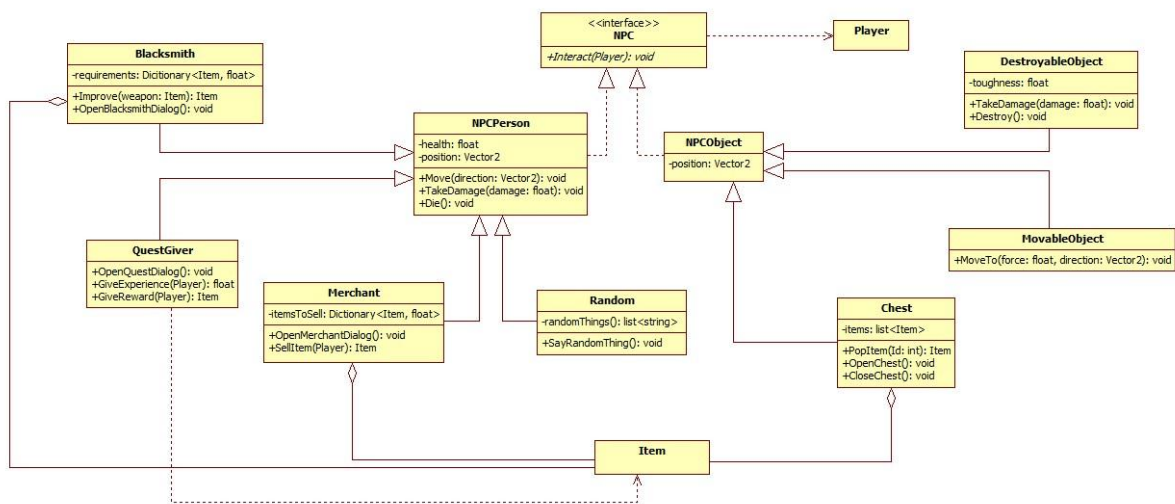
Klasa **BuilderItem** opisuje w jaki sposób gracz może wytwarzać specjalne przedmioty za pomocą receptur wykorzystując inne przedmioty.

9. Klasa Items

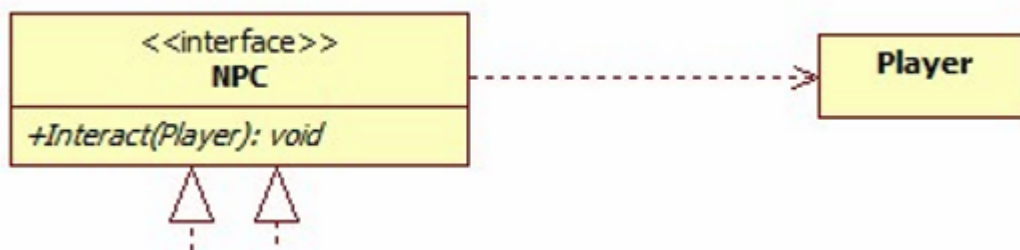


Klasa **Items** opisuje przedmioty oraz ich rodzaje.

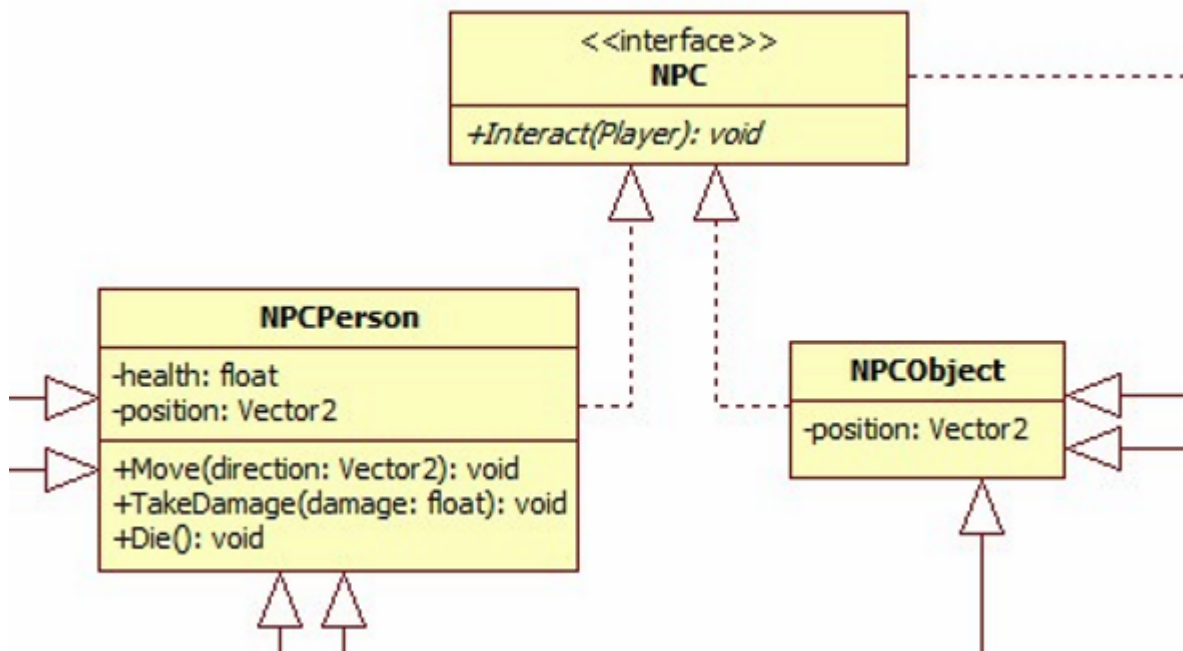
10. Klasa NPC



Na diagramie widać interfejs klasy NPC, który posiada metodę *Interact()*.

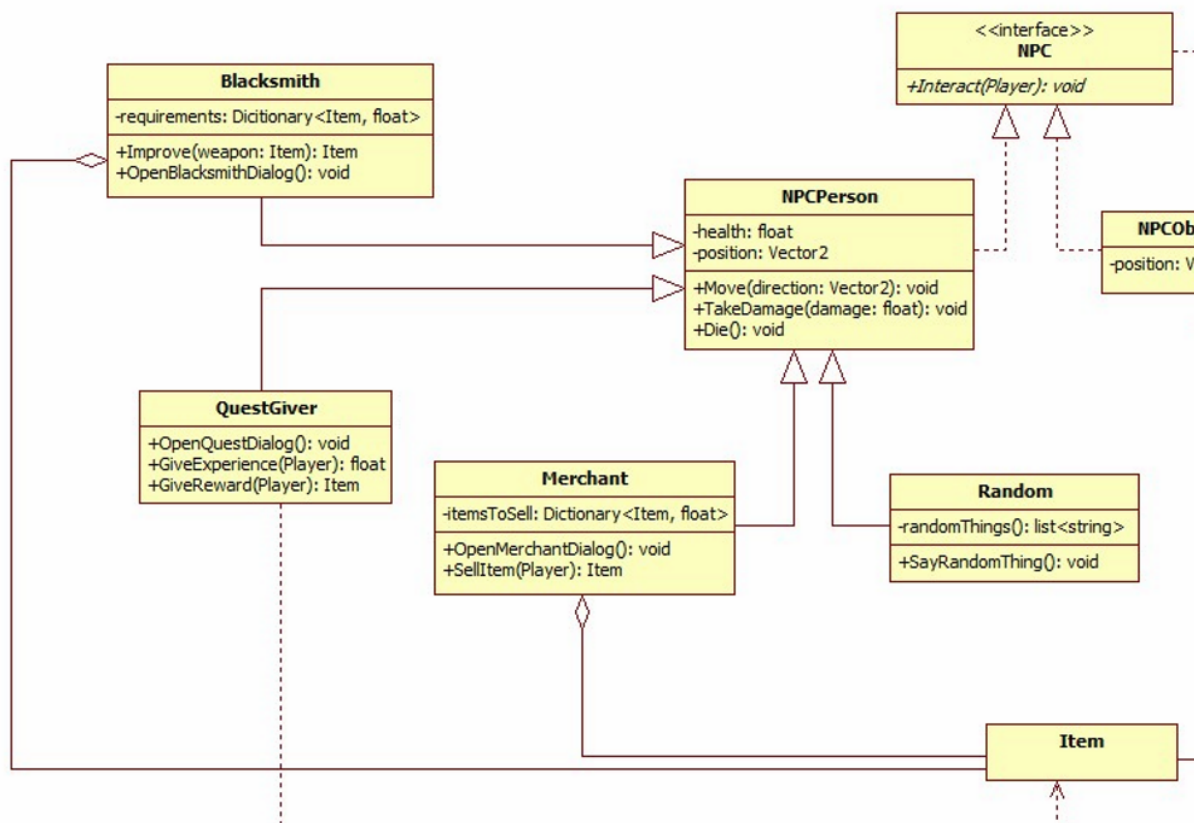


Zatem każdy obiekt będący NPC, będzie miał głównie za zadanie wejść w interakcję z graczem. Interfejs ten realizują klasy NPCPerson oraz NPCObject.



Jak łatwo się domyślić, klasy dziedziczące po NPCPerson będą postaciami osobowymi w grze, natomiast klasy dziedziczące po NPCObject będą obiektami, które gracz będzie mógł napotkać. Metoda *Interact()* jest zatem implementowana inaczej w każdej z tych najbardziej szczegółowych klas.

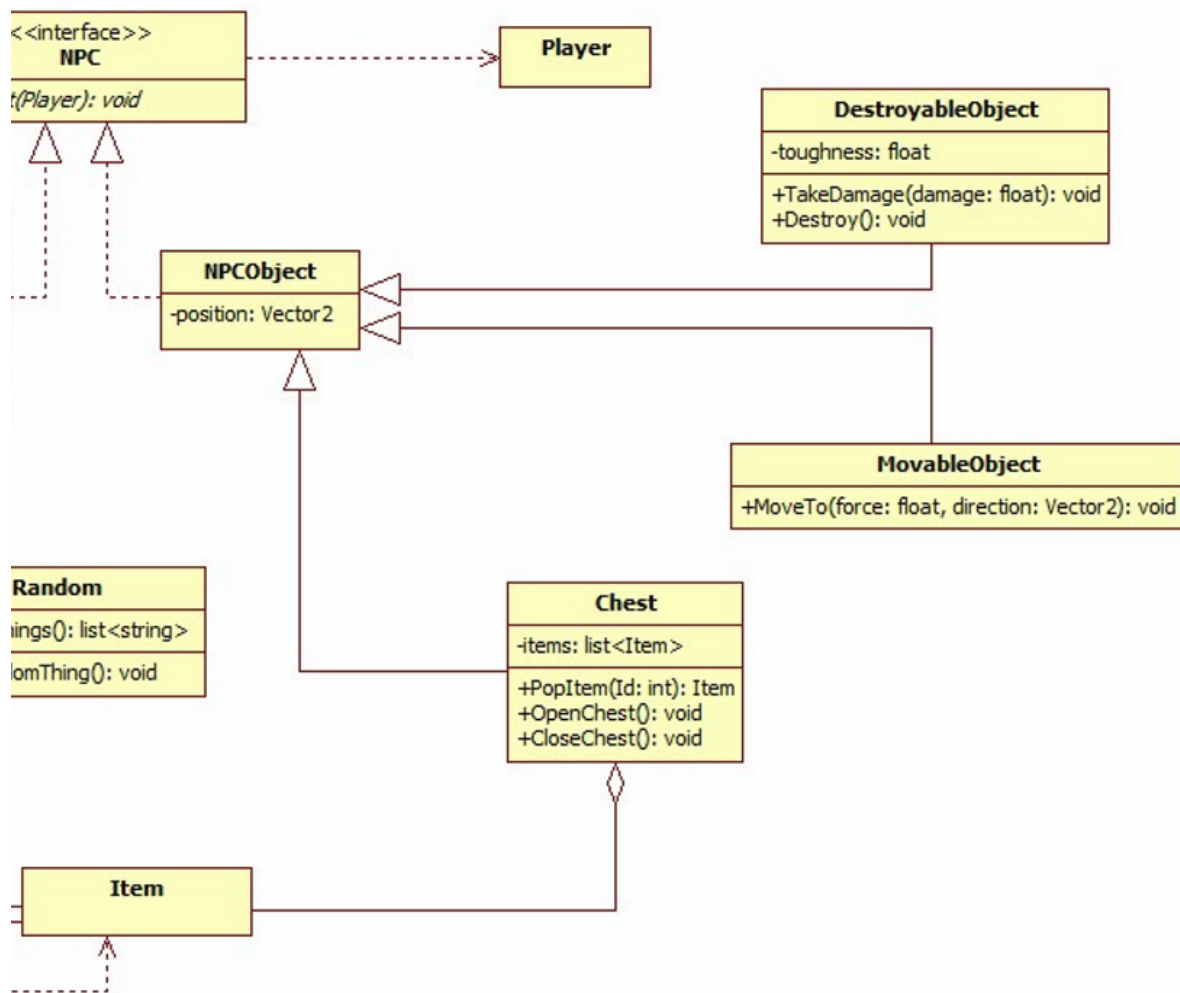
Klasy NPCPerson:



- Blacksmith - obiekt tej klasy będzie miał za zadanie ulepszać (czyli zwiększać wartości statystyk) obiektom klasy Item będącymi bronią, stąd posiada także atrybut `Dictionary<Item, float> requirements`, będący wymaganiami dla ulepszenia danej broni.
- QuestGiver - klasa, której obiekt będzie mógł udzielać graczowi zadania do wykonania, wyświetlane w operacji `OpenQuestDialog()`. Po ich ukończeniu, gracz otrzyma stosowne nagrody
- Merchant - obiekt tej klasy ma za zadanie handlować z graczem przedmiotami. Będzie się to odbywało w stworzonym do tego oknie dialogowym podczas gry.
- Random - klasa, której obiekt będzie odpowiadał za ożywianie tła gry. Operacja `SayRandomThing()` jest samokomentująca się. Gdy gracz podejdzie blisko, wyświetlony zostanie konkretny tekst, jaki ten NPC w danej chwili chce powiedzieć graczowi.

Wszystkie te klasy dziedziczą po `NPCPerson`, która zawiera operacje `Move()`, `TakeDamage()`, `Die()`, odpowiadające kolejno: poruszaniu się, otrzymywaniu obrażeń oraz śmierci danego NPC.

Klasy `NPCObject`:

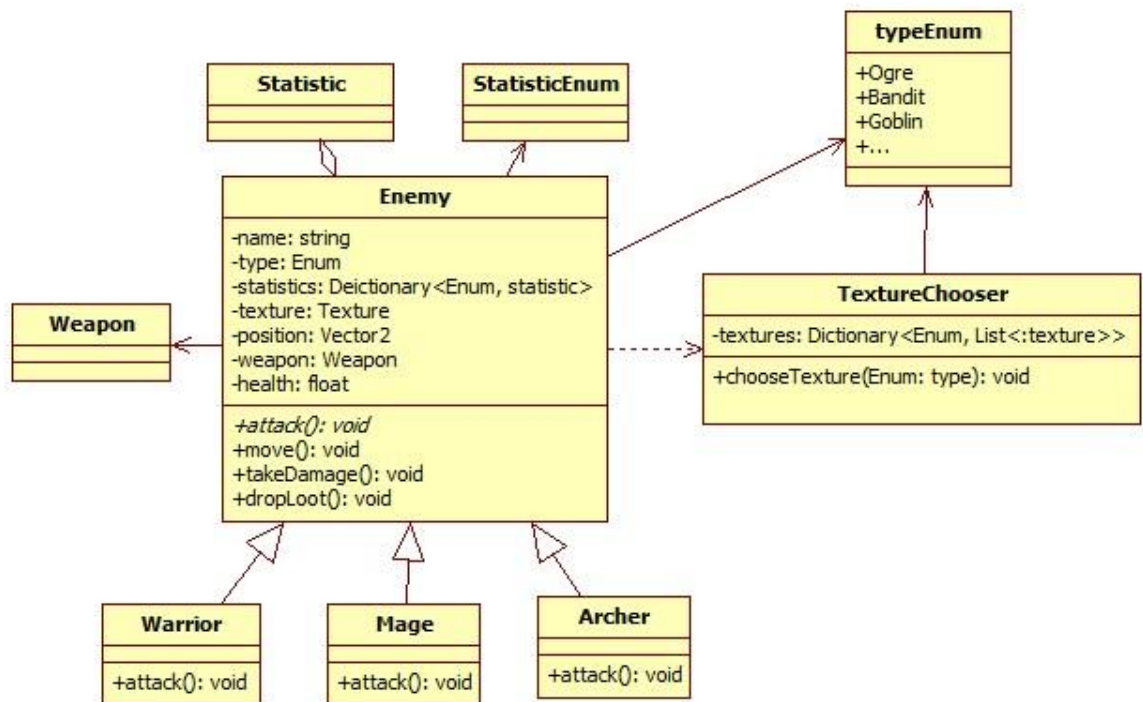


- DestroyableObject - obiekty tej klasy będą najczęściej przeszkodami blokującymi drogę gracza, a wejście w interakcję z nimi będzie oznaczać po prostu zadanie im obrażeń. Gdy atrybut toughness będzie równy lub mniejszy niż 0, obiekt będzie niszczone za pomocą operacji Destroy().
- MovableObject - obiekty tej klasy również będą przeszkodami bądź pomocami dla gracza w różnych sytuacjach, lecz nie będą mogły zostać zniszczone. Metoda MoveTo() wywoływana podczas interakcji będzie zmieniała położenie obiektu, stąd będą one "przesuwalne".
- Chest - wejście w interakcję z obiektem tej klasy spowoduje otwarcie okna dialogowego (operacja OpenChest()). Tak jak w przypadku klasy Merchant, będzie można wymieniać tu przedmioty, z tą zmianą, że nie będzie to wymagało ich zakupu.

Wszystkie te klasy dziedziczą po klasie NPCObject, która zawiera tylko atrybut position, będący położeniem obiektu w danej lokacji.

Można także zauważyć, że część klas agreguje obiekty klasy Item, a jedna klasa jest w relacji zależności do klasy Item.

11. Enemy



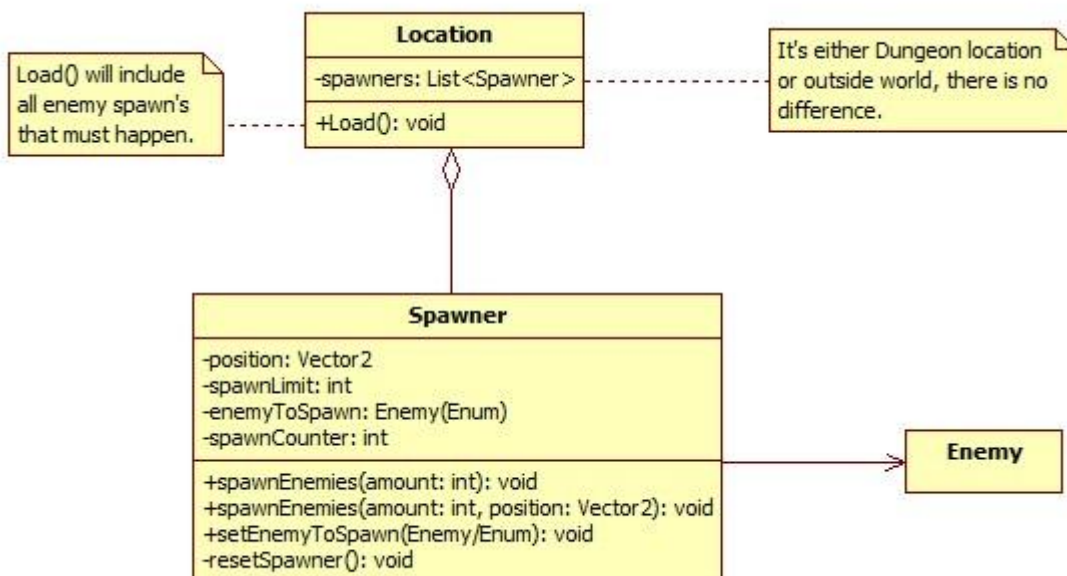
Klasa Enemy reprezentuje wroga w świecie gry. Posiada on podstawowe pola typu nazwa, typ, czy pozycja.

Tekstura wybierana jest za pomocą klasy TextureChooser, która na podstawie typu wroga, przypisuje mu teksturę.

Z klasy Enemy dziedziczą trzy klasy. Każda z nich nadpisuje abstrakcyjną klasę attack, dzięki czemu w zależności od klasy, zaatakuje mieczem, strzeli z łuku lub rzuci zaklęcie.

Każdy z wrogów ma rzeczy, które upadają na ziemię, gdy gracz go pokona.

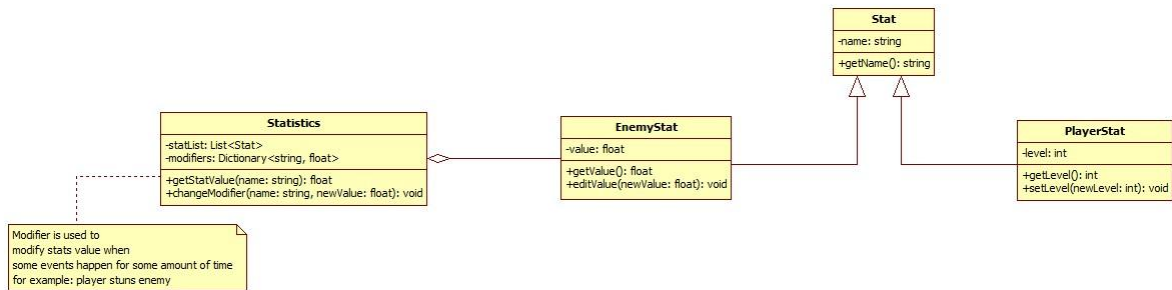
12. Klasa Spawner



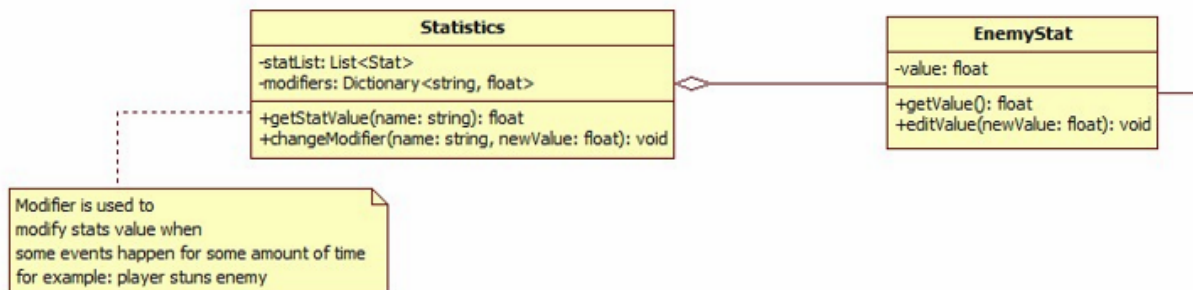
Klasa Spawner ma za zadanie tworzyć obiekty Enemy w różnych miejscach gry. Do tego służy operacja spawnEnemies(), która posiada dwa przeciążenia. Pierwsza wersja ma stworzyć taką liczbę obiektów Enemy, jaka jest podana w zmiennej amount w pozycji Spawnera. Druga wersja czyni to samo, lecz w innej, podanej do funkcji pozycji. setEnemyToSpawn() zmienia obiekt Enemy, będący w klasie, na inny, który ma być akurat

tworzony. Operacja resetSpawner() resetuje spawnCounter. Przy każdym stworzeniu obiektu typu Enemy, będzie zwiększany licznik spawnCounter, a gdy osiągnie wartość spawnLimit, już przestanie tworzyć nowe obiekty Enemy.

13. Klasa Statistics

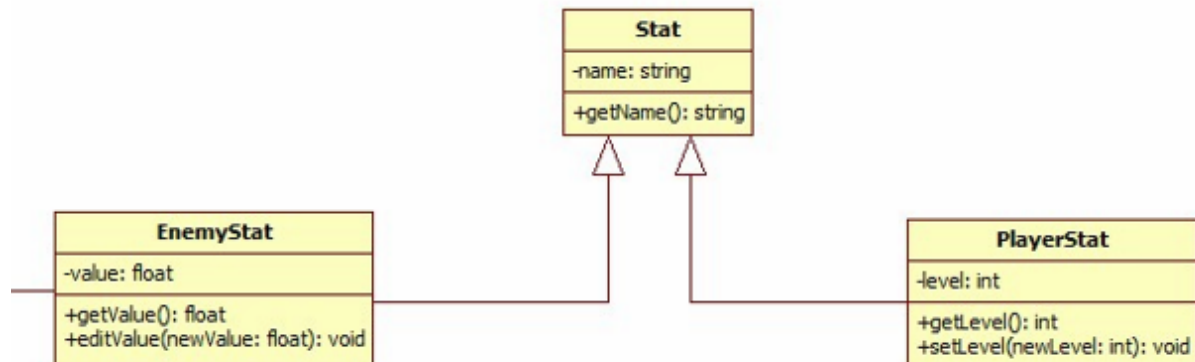


Klasa Statistics ma być pewną stałą agregującą konkretne statystyki przeciwników oraz ich mnożniki, które będą wpływać na końcową wartość danej statystyki:



Każdy obiekt przeciwnika będzie replikował tą klasę i z niej korzystał podczas rozgrywki.

Klasa stat zawiera nazwę statystyki. Dziedziczą po niej EnemyStat i PlayerStat.



Takie rozwiązanie pomaga nam podzielić statystyki na grupę dla przeciwników, oraz grupę dla gracza. Grupa dla gracza będzie wykorzystywana w klasie SkillTree i modyfikowana w inny sposób, niż statystyka przeciwnika.

Diagram sekwencji dla interakcji z NPC

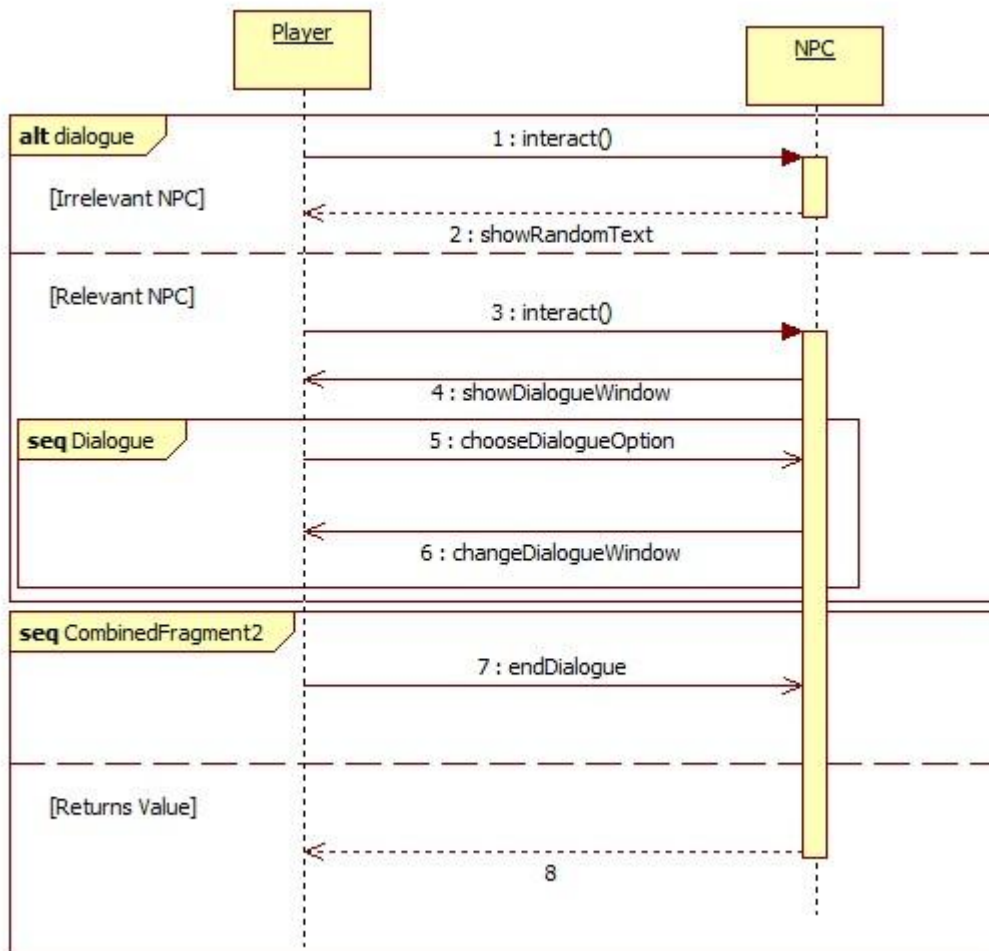
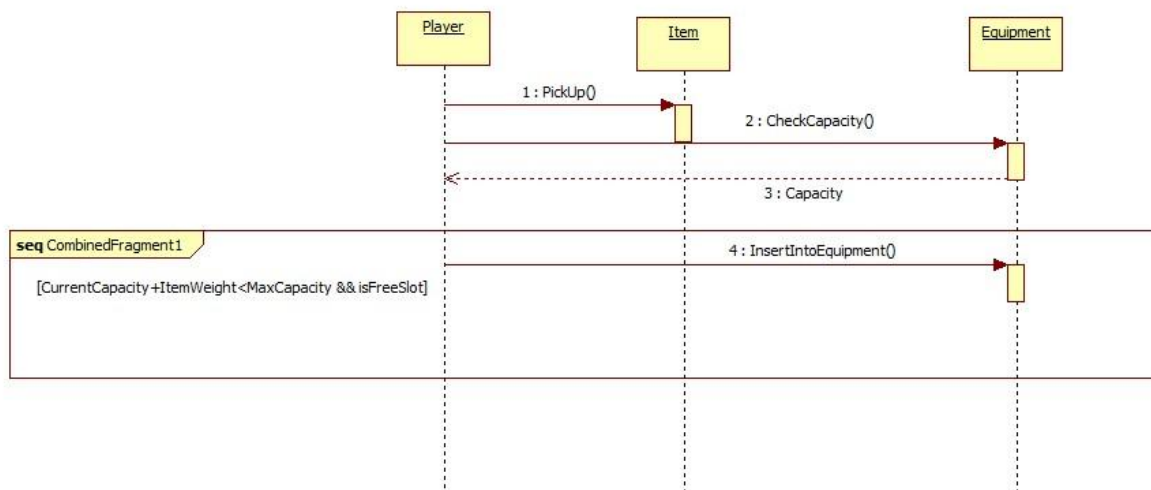
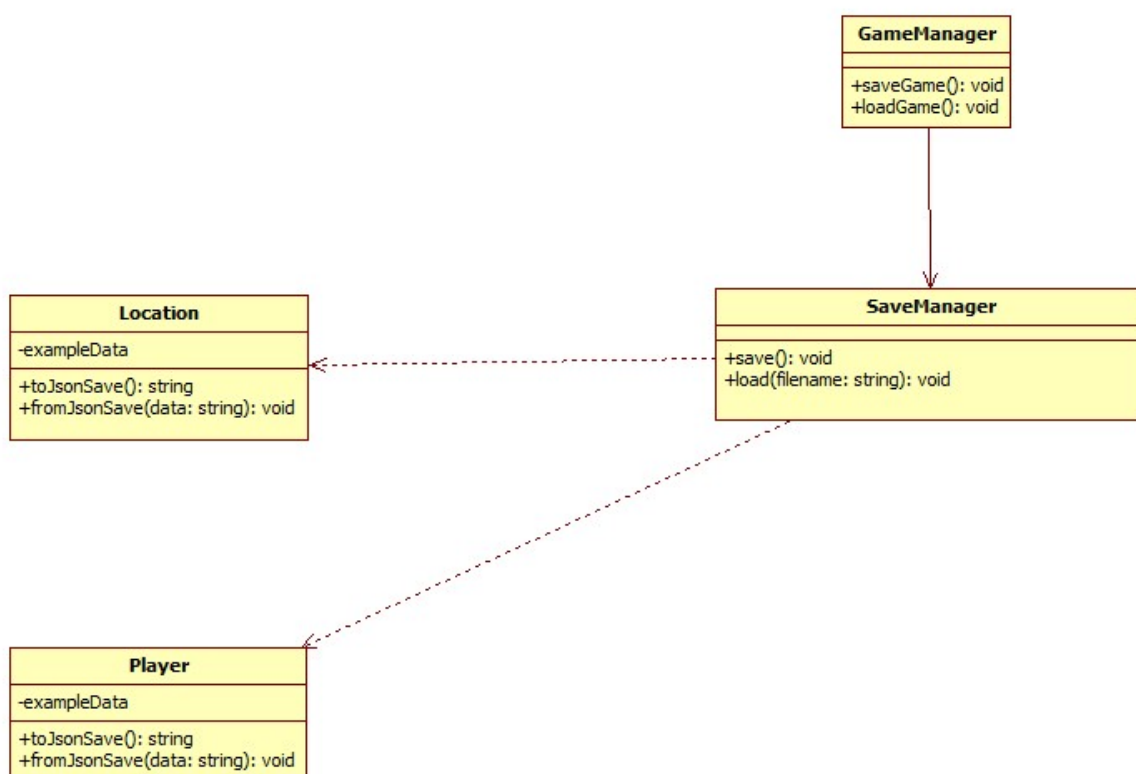


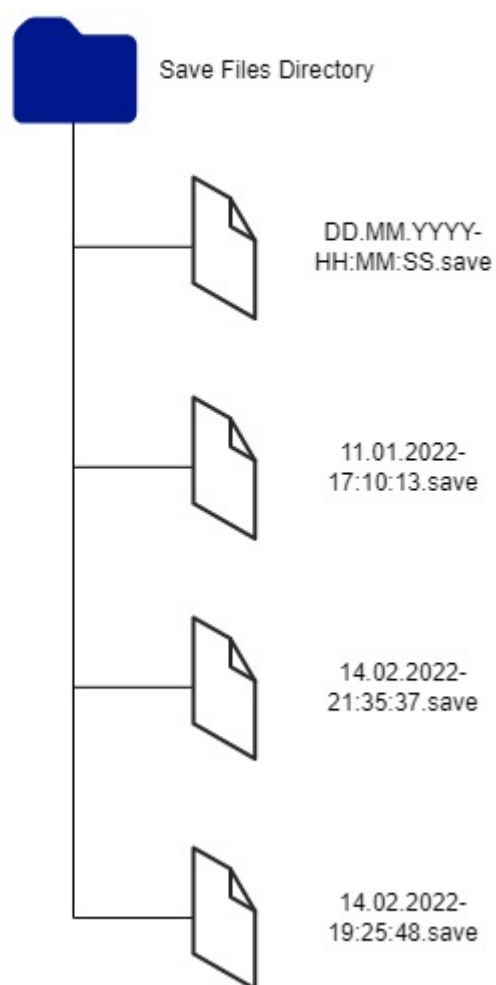
Diagram sekwencji dla interakcji z przedmiotem

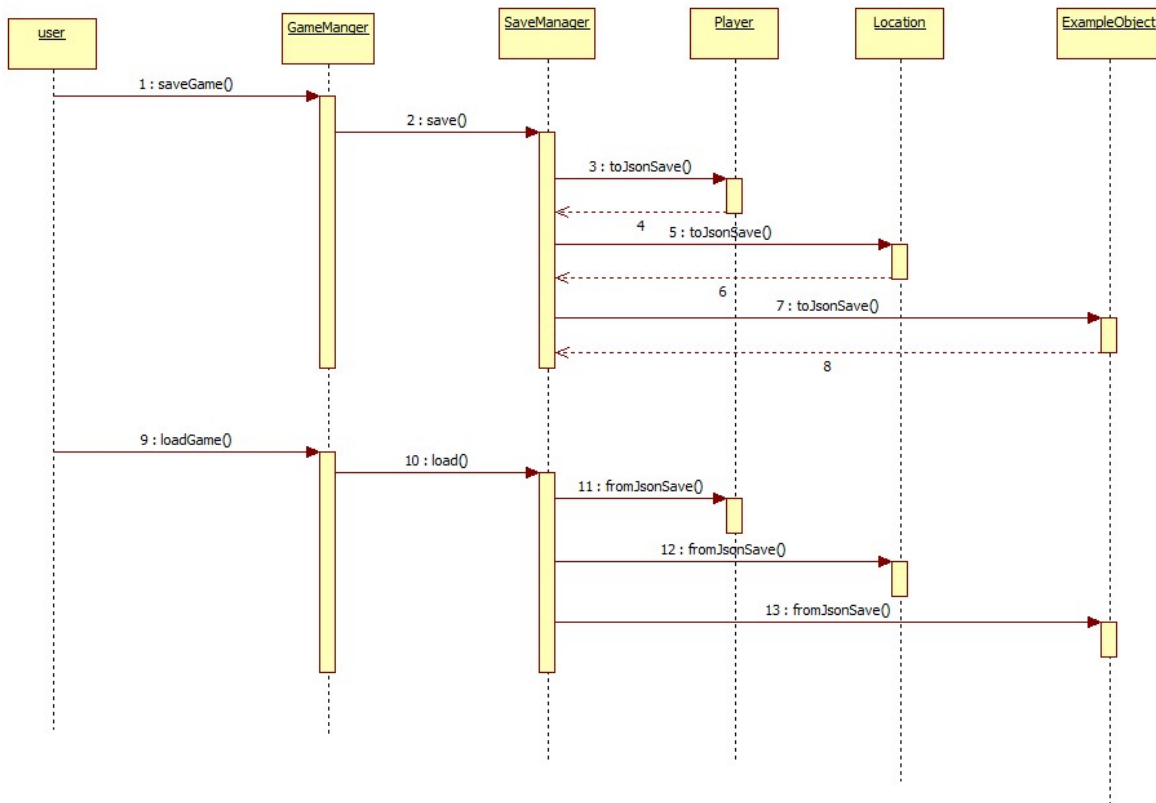


14. Klasa SaveManger oraz struktura plików zapisu.



Pliki będą zapisywane w formacie
json lecz z rozszerzeniem save.





Jeśli custom name nie jest ustawiony to save przyjmuje nazwę daty i czasu.

Cokolwiek jest zapisywane do pliku, generowane jest przez obiekt który zawiera dane do zapisu. Dany obiekt musi też wczytywać dane do niego podane.

Jeśli napotkamy strukturę danych, która nie jest obsługiwana w json (np. data, vector3 itp), definiujemy jej zapis na końcu pliku SaveFileDataStructureDefinitions.txt w katalogu /doc/Diagrams/SaveFiles grupując je złożonością. Sposób zapisu daty jest już zdefiniowany w tym pliku.

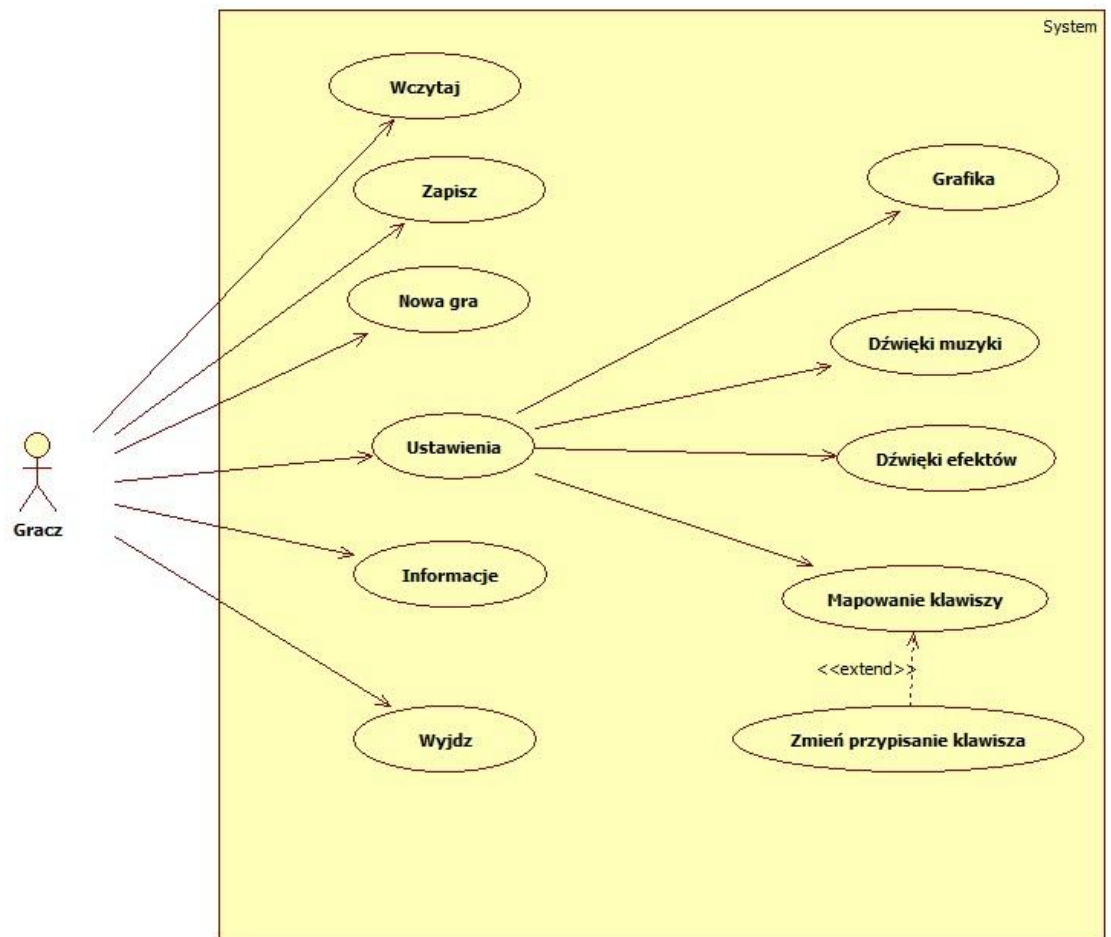
Przykładowa struktura pliku z zapisem stanu gry.

```

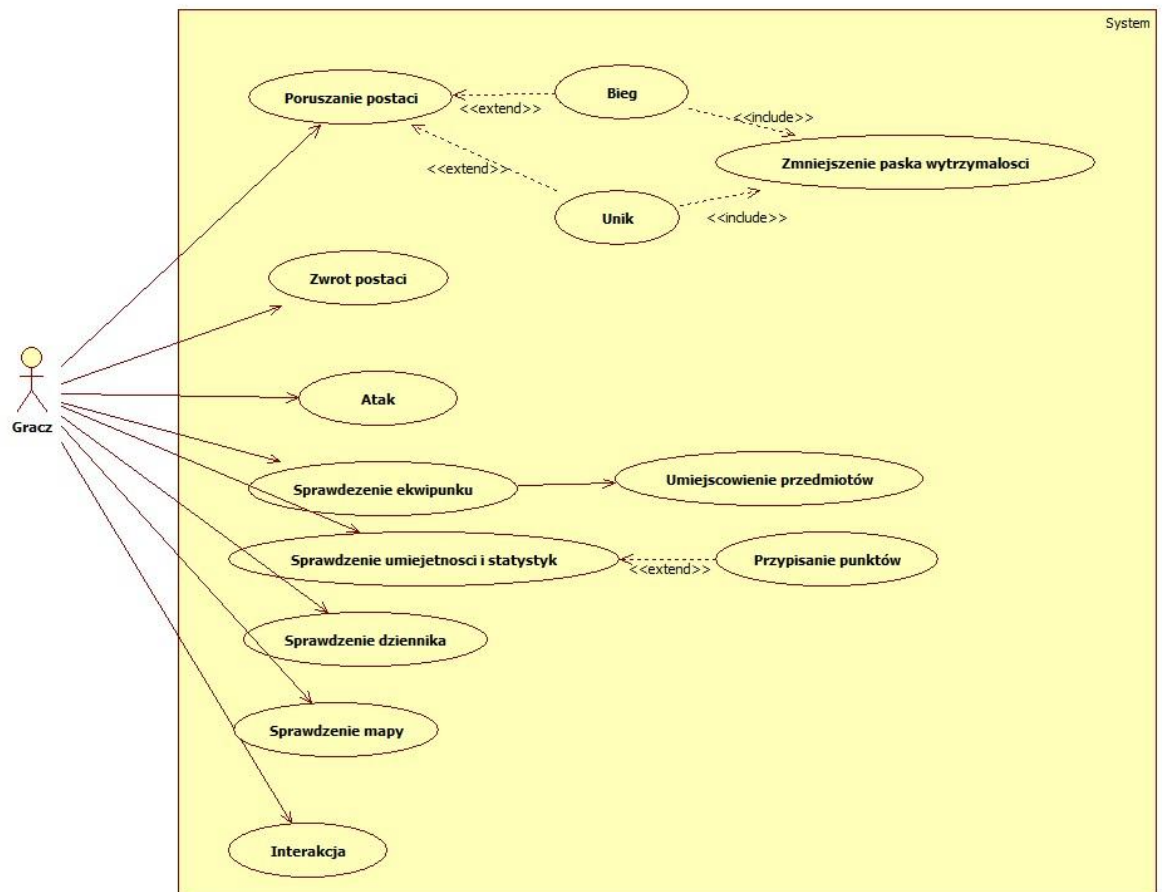
> DD.MM.YYYY-HH:MM:SS.save
{
  "date": "$DD.MM.YYYY-HH:MM:SS",
  "custom_name": "",
  "data": {
    "player_data": "$player.toJsonSave()",
    "world_data": "$world.toJsonSave()",
    "etc": "etc"
  }
}

```

1. Diagram przypadków użycia dla main menu

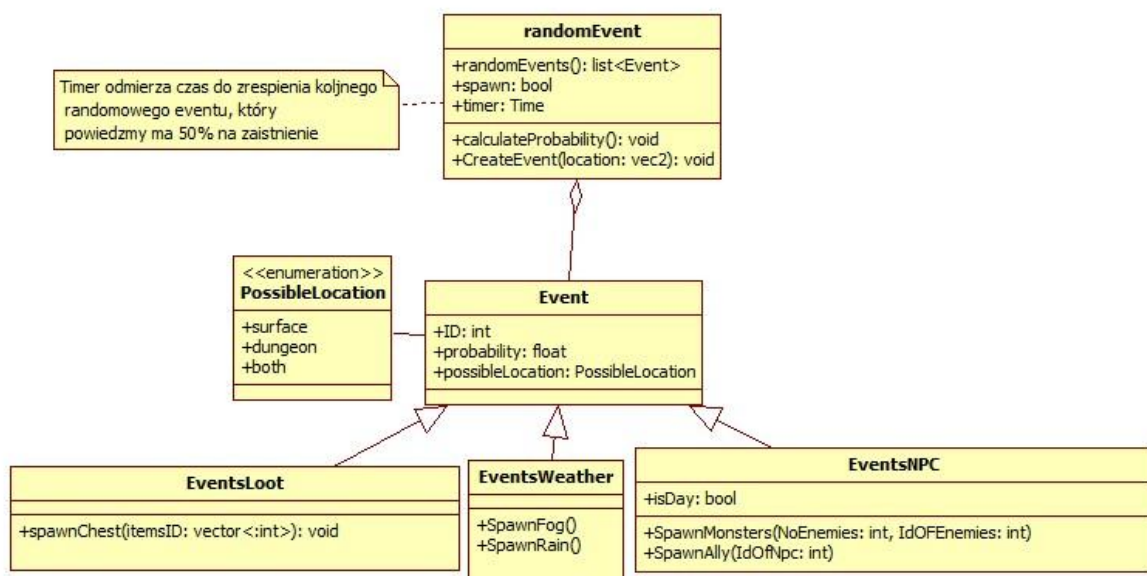


2. Diagram przypadków użycia dla gameplay'u



Nice to have:

15. Klasa random event



Klasa random event zarządza losowymi eventami, jakie mogą wystąpić w grze. Mamy enum, określający, czy event może wydarzyć się na powierzchni, w jaskini czy w obu. Eventy dziedziczą po jednej klasie i są podzielone w zależności od tego, czego dotyczą.

Scenariusz	
Warunki wstępne	
Działania	
Rezultaty	

Scenariusze testowe

Gracz - Osoba grająca w grę

Bohater - Główna postać w grze

Ruch gracza:

Scenariusz	Bohater Porusza się po lokacji
Warunki wstępne	Bohater nie jest zatrzymany przez akcje zewnętrzną
Działania	Klawisz ruchu wciśnięty przez Gracza
Rezultaty	Bohater porusza się we właściwym kierunku

Scenariusz	Bohater natrafia na przeszkodę
Warunki wstępne	Istnieje przeszkoda na drodze bohatera
Działania	Klawisz ruchu wciśnięty przez Gracza
Rezultaty	Postać gracza porusza się we wskazanym przez klawisz kierunku

Scenariusz	Gracz chce użyć sprintu
Warunki wstępne	Wartość statystyki staminy bohatera jest powyżej zera
Działania	Klawisz ruchu wciśnięty przez Gracza Gracz tym samym czasie naciska klawisz sprint
Rezultaty	Zmniejsza się wartość staminy bohatera wraz z poruszaniem się, Bohater porusza się szybciej we wskazanym kierunku

Scenariusz	Gracz chce zrobić unik
Warunki wstępne	Wartość staminy bohatera musi mieć przynajmniej wartość wymaganą do zrobienia uniku
Działania	Użytkownik naciska klawisz do poruszania się, W tym samym czasie użytkownik naciska klawisz odpowiadający za unik

Rezultaty	Zmniejsza się wartość staminy o wartość staminy konieczną do zrobienia uniku, Postać gracza robi unik we wskazanym kierunku
-----------	--

Scenariusz	Gracz spada w przepaść
Warunki wstępne	Lokacja musi zawierać przepaść
Działania	Gracz naciska klawisz poruszania się w stronę przepaści.
Rezultaty	Bohater spada w przepaść i umiera

Scenariusz	Gracz wchodzi w pułapkę
Warunki wstępne	Lokacja musi zawierać pułapkę
Działania	Gracz naciska klawisz poruszania się w stronę pułapki
Rezultaty	Bohater otrzymuje obrażenia związane z pułapką, Bohater ma mniej punktów życia lub umiera

Walka:

Scenariusz	Bohater wchodzi w pole widzenia przeciwnika
Warunki wstępne	Bohater zbliża się do przeciwnika
Działania	Gracz naciska klawisz poruszania się w stronę przeciwnika
Rezultaty	Jeśli odległość między bohaterem a przeciwnikiem jest mniejsza lub równa polu widzenia przeciwnika to przeciwnik atakuje bohatera

Scenariusz	Bohater walczy bronią wręcz
Warunki wstępne	Bohater ma wyekwipowaną broń do walki wręcz
Działania	Gracz naciska klawisz odpowiadający za zadanie ciosu
Rezultaty	Jeśli przeciwnik znajduje się w zasięgu ciosu to: Sprawdzone jest prawdopodobieństwo czy uda mu się uniknąć ciosu: Jeśli unik: Przeciwnik otrzymuje obrażenia takie, są przypisane do obiektu broni, jakiej używa bohater Jeśli nie unik: Nic się nie dzieje

Scenariusz	Bohater walczy bronią dystansową
------------	----------------------------------

Warunki wstępne	Bohater ma wyekwipowaną broń do wali dystansowej, Bohater ma amunicję, Lokacja może zawierać przeszkodę
Działania	Gracz naciska przycisk ataku
Rezultat	Bohater odwraca się w stronę kursora, Bohater strzela pociskiem w kierunku kursora
Warunek	Pocisk natrafia na przeciwnika
Rezultat	Pocisk znika lub przebija się (z szansą) a przeciwnik otrzymuje obrażenia o zadanej wartości
Warunek	Pocisk natrafia na przeszkodę
Rezultat	Pocisk znika
Warunek	Pocisk leci swobodnie
Rezultat	Pocisk leci poza widzialny przez Gracza obszar i znika.

Scenariusz	Bohater walczy bronią magiczną
Warunki wstępne	Bohater ma wyekwipowaną broń magiczną, Bohater ma więcej niż 0 punktów many Lokacja może zawierać przeszkodę
Działania	Gracz naciska przycisk ataku.
Rezultaty	Bohater odwraca się w stronę kursora Bohater używa broni magicznej tworząc pocisk w kierunku kursora
Warunek	Pocisk trafia w przeciwnika
Rezultat True	Przeciwnik otrzymuje obrażenia o wartości jakie zadaje broń magiczna lub zmniejszonej, jeżeli posiada odporność na magię
Rezultat False Warunek	Pocisk trafia w przeszkodę
Rezultat True	Pocisk znika lub przechodzi przez przeszkodę jeśli broń ma taką właściwość
Rezultat False	Pocisk dolatuje do momentu aż osiągnie maksymalny zasięg, spada na ziemię i znika

Scenariusz	Bohater umiera w walce
Warunki wstępne	W lokacji przebywają przeciwnicy
Działania	Przeciwnik zada bohaterowi cios, po którym HP gracza będzie mniejsze lub równe 0
Rezultaty	Bohater wykonuje animacje śmierci Włącza się menu

Scenariusz	Bohater unika ciosu
Warunki wstępne	Bohater ma wystarczająco staminy W lokacji przebywają przeciwnicy
Działania	Przeciwnik zadaje cios bohaterowi W tym samym czasie gracz naciska klawisz odpowiedzialny za unik
Rezultaty	Bohater nie otrzymuje obrażeń podczas uniku

Umiejętności

Scenariusz	Zwiększenie statystyki
Warunki wstępne	
Działania	
Rezultaty	

Scenariusz	Użycie umiejętności
Warunki wstępne	Umiejętność jest odblokowana
Działania	Gracz otwiera okno drzewka umiejętności Gracz przeciąga odblokowaną umiejętność do wolnego slotu wywołania umiejętności Umiejętność jest "w użyciu" pod klawiszem slotu Gracz naciska odpowiedni klawisz
Rezultaty	Umiejętność jest wykonana

Scenariusz	Odblokowanie umiejętności
Warunki wstępne	Poziom statystyki danej umiejętności jest wystarczający
Działania	Gracz otwiera okno drzewka umiejętności Najeżdża kursorem na daną umiejętność i naciska by odblokować
Rezultaty	Umiejętność jest odblokowana i dostępna do użycia

Scenariusz	Bohater ma w użyciu umiejętność <i>Furii</i>
------------	--

Warunki wstępne	Umiejętność <i>Furii</i> jest odblokowana Otwarte jest okno drzewka umiejętności
Działania	Gracz przeciąga umiejętność <i>Furii</i> do miejsca umiejętności "w użyciu"
Rezultaty	Statystyki szybkości poruszania się i zadawanych obrażeń zostają zwiększone

Scenariusz	Bohater używa umiejętności <i>Wir Miecza</i>
Warunki wstępne	Umiejętność <i>Wir Miecza</i> jest odblokowana Umiejętność jest "w użyciu" pod jednym z klawiszy Użycie umiejętności nie jest zablokowane przez opóźnienie (cooldown)
Działania	Gracz naciska klawisz pod którym jest dana umiejętność
Rezultaty	Bohater kręci się dookoła z bronią Każdy przeciwnik w zasięgu otrzymuje obrażenia Na daną umiejętność nakładane jest opóźnienie

Scenariusz	Bohater ma w użyciu umiejętność <i>Zatrucia</i>
Warunki wstępne	Umiejętność <i>Zatrucie</i> jest odblokowana Otwarte jest okno drzewka umiejętności
Działania	Gracz przeciąga umiejętność <i>Zatrucie</i> do miejsca umiejętności "w użyciu"
Rezultaty	Każdy pocisk z broni dystansowej otrzymuje efekt zatrucia Przeciwnik trafiony tym pociskiem przez kilka sekund traci po kilka HP

Scenariusz	Bohater używa umiejętności <i>Uniku</i>
Warunki wstępne	Umiejętność <i>Uniku</i> jest odblokowana Umiejętność jest "w użyciu" pod jednym z klawiszy Użycie umiejętności nie jest zablokowane przez opóźnienie (cooldown)
Działania	Gracz naciska klawisz pod którym jest dana umiejętność
Rezultaty	Bohater staje się na kilka sekund odporny na wszystkie możliwe obrażenia

Scenariusz	Bohater ma w użyciu umiejętność <i>Laseru</i>
Warunki wstępne	Umiejętność <i>Laser</i> jest odblokowana Otwarte jest okno drzewka umiejętności
Działania	Gracz przeciąga umiejętność <i>Laseru</i> do miejsca umiejętności "w użyciu"
Rezultaty	Każde użycie broni magicznej zamiast wypuszczać pociski, staje się teraz wiązką laseru

Scenariusz	Bohater używa umiejętności <i>Regeneracji</i>
Warunki wstępne	Umiejętność <i>Regeneracji</i> jest odblokowana Umiejętność jest "w użyciu" pod jednym z klawiszy Użycie umiejętności nie jest zablokowane przez opóźnienie (cooldown) Wartość punktów many jest większa od zera

Działania	Gracz naciska klawisz pod którym jest dana umiejętność
Rezultaty	W czasie gdy klawisz jest wciśnięty, bohater traci punkty many, a w zamian otrzymuje punkty HP

Przedmioty

Scenariusz	Sprawdzenie przedmiotu
Warunki wstępne	Gracz ma otwarty ekwipunek Gracz ma przedmiot w ekwipunku
Działania	Gracz najeżdża kursorem na przedmiot
Rezultaty	Gracz może odczytać nazwę przedmiotu Gracz może odczytać statystyki przedmiotu Gracz może odczytać opis przedmiotu Gracz może odczytać przeznaczenie przedmiotu

Interakcja z przedmiotem

Scenariusz	Gracz zakłada przedmiot na wyposażenie
Warunki wstępne	Bohater ma otwarty ekwipunek Bohater posiada przedmiot do założenia
Działania	Gracz przeciąga przedmiot na wyposażenie
Rezultaty	Bohater założył przedmiot Jeśli Bohater miał już nałożony przedmiot, następuje podmienienie przedmiotów Jeśli Bohater nie spełnia wymagań przedmiotu, wyświetla się komunikat że gracz nie spełnia wymagań i przedmiot wraca do ekwipunku

Scenariusz	Gracz przekłada przedmioty w plecaku
Warunki wstępne	Ekwipunek jest otwarty Bohater posiada przedmiot
Działania	Gracz przeciąga przedmiot z jednego slotu na inny
Rezultaty	Gracz może przekładać przedmioty w ekwipunku

Scenariusz	Użycie przedmiotu
Warunki wstępne	Ekwipunek jest otwarty Bohater posiada przedmiot do użycia
Działania	Gracz naciska dwukrotnie kursorem na przedmiot

Rezultaty	<p>Bohater użył przedmiot, jego statystyki zmodyfikowały się</p> <p>Bohater nie użył przedmiotu, ponieważ nie spełnia wymagań przedmiotu, wyświetla się komunikat że gracz nie spełnia wymagań i przedmiot wraca na miejsce</p>
-----------	---

Scenariusz	Gracz zdejmuję przedmiot z wyposażenia
Warunki wstępne	Ekwipunek jest otwarty
Działania	Gracz przeciąga przedmiot z wyposażenia do ekwipunku
Rezultaty	<p>Bohater zdejmuję przedmiot z wyposażenia</p> <p>Bohater nie może zdjąć przedmiotu, ponieważ nie ma miejsca w ekwipunku, wyświetla się komunikat że ekwipunek jest pełny</p>

Scenariusz	Gracz usuwa przedmiot z ekwipunku
Warunki wstępne	Ekwipunek jest otwarty
Działania	Gracz przeciąga przedmiot poza ekwipunek
Rezultaty	Wyświetla się komunikat czy jesteś pewny że chcesz wyrzucić przedmiot

Scenariusz	Gracz tworzy nowy przedmiot za pomocą buildera
Warunki wstępne	<p>Ekwipunek jest otwarty</p> <p>Bohater posiada przedmiotu do craftingu</p>
Działania	Gracz przeciąga przedmioty na siebie.
Rezultaty	<p>Bohater otrzymał nowy przedmiot</p> <p>Bohater nie otrzymał przedmiotu, ponieważ nie posiadał wystarczającej ilości składników, przedmioty wracają do ekwipunku</p> <p>Bohater niepoprawnie połączył przedmioty, przedmioty wracają do ekwipunku</p> <p>Bohater nie otrzymał przedmiotu ponieważ nie ma miejsca w ekwipunku</p>

Interakcja z NPC

Scenariusz	Interakcja z NPC
Działania	<p>Bohater podchodzi na bliską odległość do NPC</p> <p>Najeżdża kursorem na NPC i naciska klawisz interakcji</p>
Warunek:	NPC jest istotny
Rezultaty True	Otwiera się okno dialogowe

Rezultaty False	Nad NPC wyświetla się chmurka z randomowym tekstem i znika po kilku sekundach
--------------------	---

Scenariusz	Interakcja z obiektem, który da się zniszczyć
Warunki wstępne	Bohater znajduje się w bliskiej odległości obiektu
Działania	Bohater podchodzi na bliską odległość do NPC Najeżdża kursorem na NPC i naciska klawisz interakcji
Warunek	Bohater posiada przedmiot, dzięki któremu można zniszczyć obiekt (czasami żaden nie jest wymagany)
Rezultaty True	Obiekt jest zniszczony, odgrywa się animacja jego rozpadu/zniszczenia i znika
Rezultaty False	Nic się nie dzieje

Scenariusz	Interakcja z obiektem, który da się przesunąć
Warunki wstępne	Bohater znajduje się w bliskiej odległości obiektu
Działania	Bohater podchodzi na bliską odległość do NPC Najeżdża kursorem na NPC i naciska klawisz interakcji
Rezultaty	Obiekt przesuwa się w tym kierunku, w którym do niego zwrócony jest bohater

NPC?

Transakcje z NPC

Scenariusz	Bohater kupuje przedmiot u handlarza
Warunki wstępne	Bohater posiada wolne miejsce w ekwipunku Otwarte okno dialogowe handlu
Działania	Gracz przytrzymuje klawiszem myszki przedmiot znajdujący się w ekwipunku handlarza i przeciąga do swojego ekwipunku
Warunek	Bohater posiada odpowiednią ilość pieniędzy w ekwipunku i aktualny ciężar gracza jest mniejszy niż maksymalny
Rezultaty True	Przedmiot zostaje w ekwipunku bohatera Odpowiednia suma pieniędzy jest graczowi odjęta, a dana handlarzowi
Rezultaty False	Przedmiot zostaje w ekwipunku handlarza Wyświetla się komunikat, że brakuje bohaterowi pieniędzy lub nie może

	unieść więcej przedmiotów
--	---------------------------

Scenariusz	Bohater sprzedaje przedmiot handlarzowi
Warunki wstępne	Otwarte okno dialogowe handlu
Działania	Gracz przytrzymuje klawiszem myszki przedmiot znajdujący się w ekwipunku gracza i przeciąga do ekwipunku handlarza
Warunek	Handlarz posiada konieczną ilość pieniędzy
Rezultaty Posiada	Przedmiot zostaje w ekwipunku handlarza Odpowiednia suma pieniędzy jest odjęta handlarzowi, a dana graczowi Zmniejsza się aktualny ciężar, jaki gracz nosi
Rezultaty	Przedmiot zostaje w ekwipunku bohatera Wyświetla się komunikat, że brakuje handlarzowi pieniędzy

Scenariusz	Bohater otrzymuje zadanie od NPC z zadaniami
Warunki wstępne	Otwarte okno dialogowe z NPC dającym zadania
Działania	Wyświetlone są nazwy zadań: <i>wykonane / do wykonania</i> Gracz wybiera możliwe zadanie do wykonania Po naciśnięciu na nazwę zadania, wyświetla się jego opis (co trzeba wykonać) oraz nagroda za wykonanie Gracz naciska przycisk <i>wykonaj zadanie</i>
Rezultaty	Do dziennika zostało dodane zadanie do wykonania wraz z opisem i nagrodą

Scenariusz	Bohater kończy zadanie
Warunki wstępne	Zadanie zostało wykonane (Jest taka informacja w dzienniku) Otwarte jest okno dialogowe z NPC dającym zadania
Działania	Wyświetlone są nazwy zadań: <i>wykonane / do wykonania</i> Przy zadaniu aktualnie wykonywanym (a już faktycznie ukończonym) pokazuje się przycisk <i>zakończ zadanie</i> Gracz go naciska
Warunek	Bohater posiada wolne miejsce w ekwipunku i ciężar wymagany do uniesienia przedmiotów
Rezultaty True	Zadanie przechodzi do zakładki <i>wykonane</i> Bohater otrzymuje związaną z nim nagrodę
Rezultaty False	Zadanie przechodzi do zakładki <i>wykonane</i> Jeśli nagroda jest przedmiotem: Przedmiot zostaje na ziemi Jeśli nie jest: Bohater otrzymuje nagrodę

Scenariusz	Bohater ulepsza przedmiot u kowala
Warunki wstępne	Otwarte jest okno dialogowe z kowalem
Działania	Gracz przeciąga przedmiot do slotu ulepszenia przedmiotów Wyświetlana jest ilość pieniędzy jaką bohater musi zapłacić lub dodatkowy przedmiot jaki musi posiadać Gracz naciska przycisk ulepsz
Warunek	Bohater spełnia wymagania do ulepszenia
Rezultaty True	Przedmiot zostaje ulepszony (zwiększone są jego statystyki) Wymagane przedmioty znikają z ekwipunku bohatera Odpowiednia ilość pieniędzy jest odjęta bohaterowi Przedmiot pojawia się w pierwszym wolnym slotcie bohatera
Rezultaty False	Pojawia się komunikat, czego brakuje do ulepszenia Przedmiot pojawia się w pierwszym wolnym slotcie bohatera

Scenariusz	Gracz wymienia (wyjmuje lub zostawia) przedmioty w skrzyni
Warunki wstępne	Otwarte jest okno dialogowe ze skrzynią
Działania	Gracz przeciąga przedmioty ze skrzyni do ekwipunku bohatera lub odwrotnie
Warunek	Bohater posiada ciężar wymagany do uniesienia przedmiotów
Rezultaty True	Przedmiot pojawia się w odpowiednim slotcie skrzyni lub bohatera Jeśli przedmiot był przenoszony z ekwipunku bohatera do skrzyni, zmniejsza się aktualny ciężar, jaki gracz nosi
Rezultaty False	Przedmiot zostaje w skrzyni, jeśli był przenoszony ze skrzyni do ekwipunku bohatera

Zmiana lokacji

Scenariusz	Zmiana lokacji
Warunki wstępne	Dungeon jest odblokowany
Działania	Gracz przechodzi przez wejście dungeonu
Rezultaty	Gracz znalazł się w dungeonie

Minimapa

Scenariusz	Sprawdzenie nazw lokacji
Warunki wstępne	Otwarta minimapa
Działania	Gracz najeżdża kursorem na minimapę
Rezultaty	Wyświetla się nazwa lokacji

Scenariusz	Sprawdzenie opisu lokacji
Warunki wstępne	Otwarta minimapa
Działania	Gracz naciska przycisk myszki pod nazwa lokacji
Rezultaty	Wyświetla się opis lokacji

Dziennik

Scenariusz	Otwarcie dziennika
Działania	Gracz naciska przycisk odpowiedzialny za wyświetlanie dziennika
Rezultaty	Dziennik wyświetla się na ekranie

Scenariusz	Sprawdzenie glosariusza, bestiariusza lub zadań
Warunki wstępne	Dziennik jest otwarty
Działania	Gracz naciska zakładkę bestiariusza, glosariusza lub zadań
Rezultaty	Dziennik wyświetla bestiariusz, glosariusz lub zadania

Scenariusz	Sprawdzenie informacji o konkretnej postaci
Warunki wstępne	Dziennik jest otwarty, Zakładka glosariusza jest aktywna
Działania	Gracz klika na konkretną postać w spisie postaci
Rezultaty	Informacje o postaci zostają wyświetlone.

Scenariusz	Sprawdzenie informacji o konkretnym potworze
Warunki wstępne	Dziennik jest otwarty, Zakładka bestiariusza jest aktywna

Działania	Gracz klika na konkretnego potwora w spisie potworów
Rezultaty	Informacje o potworze zostają wyświetlone

Scenariusz	Sprawdzenie informacji o konkretnym zadaniu
Warunki wstępne	Dziennik jest otwarty, Zakładka zadań jest aktywna
Działania	Gracz klika na konkretną postać w spisie postaci
Rezultaty	Informacje o zadaniu zostaną wyświetlone

Scenariusz	Zamknięcie dziennika
Warunki wstępne	Dziennik jest otwarty
Działania	Gracz klika ikonę zamknięcia dziennika
Rezultaty	Dziennik znika z ekranu

Wrogowie

Scenariusz	Wejście w pole widzenia wroga
Warunki wstępne	Wróg jest w pobliżu
Działania	Bohater wchodzi w pole widzenia wroga
Rezultaty: Mag, Łucznik	Wróg oddala się od bohatera Wróg atakuje bohatera
Rezultaty: Wojownik	Wróg podchodzi do bohatera Wróg atakuje bohatera

Scenariusz	Zadanie obrażeń
Warunki wstępne	Wróg zauważył bohatera
Działania	Wróg zaatakował
Rezultaty: Trafienie	Wróg odgrywa animację ataku Bohater otrzymuje obrażenia zgodnie ze statystykami
Rezultaty: Chybienie	Wróg odgrywa animację ataku

Scenariusz	Przyjmowanie obrażeń
Warunki wstępne	Gracz wcisnął przycisk ataku
Działania	Bohater zaatakował
Rezultaty	Wróg otrzymuje obrażenia zgodnie z przelicznikami

Scenariusz	Śmierć wroga
Warunki wstępne	Gracz zadaje przeciwnikowi obrażenia
Działania	Liczba punktów zdrowia wroga spadła do zera lub mniej
Rezultaty	Wróg umiera a jego postać odgrywa animację śmierci i znika Wróg upuszcza łupy w zależności od prawdopodobieństwa na upuszczenie łupu Gracz otrzymuje punkty doświadczenia

Administracja czyli menu, ustawienia, opcje

Zapis gry

Scenariusz	Włączenie okna zapisu gry
Warunki wstępne	Bohater nie uczestniczy w walce Dziennik jest włączony
Działania	Gracz klika ikonę zapisu
Rezultaty	Dziennik wyświetla okno zapisu gry

Scenariusz	Zapisanie stanu gry na słocie
Warunki wstępne	Okno zapisu/wczytania gry jest aktywne
Działania	Gracz klika na slot zapisu Gracz wpisuje nazwę zapisu Gracz klika przycisk "zapisz"
Rezultaty:	Potrzebne dane zostają zczytane z gry

brak zapisu	Wszystkie dane zostają zapisane do pliku Json Zapis pojawia się na wybranym wcześniej slotcie
Rezultaty: Nadpisanie zapisu	Potrzebne dane zostają zczytane z gry Wszystkie dane zostają nadpisane w pliku Json

Wczytywanie gry

Scenariusz	Włączenie okna wczytania podczas gry
Warunki wstępne	Dziennik jest włączony
Działania	Gracz kilka ikonę wczytania LUB Bohater ginie
Rezultaty	Dziennik wyświetla okno wczytania gry

Scenariusz	Wczytanie gry
Warunki wstępne	Okno wczytania gry jest otwarte LUB Okno wczytania gry jest otwarte w main menu
Działania	Gracz wybrał zapis Gracz kliknął przycisk "wczytaj"
Rezultaty	Dane z zapisu zostają załadowane do gry

Zmiana ustawień

Scenariusz	Otwarcie menu ustawień
Warunki wstępne	Menu główne jest otwarte
Działania	Gracz klika przycisk ustawień
Rezultaty	Okno ustawień otwiera się

Scenariusz	Zmiana głośności
Warunki wstępne	Okno ustawień jest otwarte
Działania	Gracz przesuwają suwak głośności

Rezultaty	Głośność muzyki i efektów dźwiękowych zmienia się adekwatnie do ustawienia
-----------	--

Scenariusz	Zmiana jasności
Warunki wstępne	Okno ustawień jest otwarte
Działania	Gracz przesuwając suwak jasności
Rezultaty	Jasność zmienia się adekwatnie do ustawienia

Scenariusz	Otwarcie okna mapowania klawiszy
Warunki wstępne	Okno ustawień jest otwarte
Działania	Gracz klika przycisk mapowania klawiszy
Rezultaty	Okno mapowania klawiszy otwiera się

Scenariusz	Zmiana mapowania klawiszy
Warunki wstępne	Okno mapowania klawiszy jest otwarte
Działania	Gracz klika klawisz przypisany do danej akcji Gracz klika klawisz na klawiaturze
Rezultaty	Klawisz przypisany do danej akcji zostaje zmieniony na ten wciśnięty przez gracza