AGH Kraków
Karol Haiduk
Magdalena Szkółka
Sylwester Tomaszewski

Rogalik
Dokumentacja architektury systemu

Wersja 1.0

Historia zmian

Data	Wersja	Opis	Autor
26/10/2012	1.0	Pierwsza wersja dokumentu.	Karol Haiduk
30/12/2012	2.0	Uzupełnienie architektury	Karol Haiduk
5.01.2013	3,	Uzupełnienie architektury	Sylwester Tomaszewski

Spis treści

Dokumentacja architektury systemu

1. Wprowadzenie

1.1 Cel

Ten dokument opisuje architekturę tworzonego projektu, zawarte w nim klasy z uwzględnieniem procedur i zmiennych.

1.2 Zakres

Dokument ten dotyczy projektu utworzenia gry RPG na zaliczenie przedmiotu Inżynieria Oprogramowania.

1.3 Pojęcia

Pojęcia zawarte w Dokumentacji architektury systemu wyjaśnione są w Słowniku.

1.4 Odniesienia

Słownik, zawarty w pliku 02_slownik.pdf

1.5 Streszczenie

Reszta dokumentu zawiera kolejno: reprezentację architektury systemu, jej cele i ograniczenia, perspektywę przypadków użycia, perspektywę logiczną (opisującą pakiety i klasy), opis interfejsu oraz omówienie rozmiaru, wydajności i jakości gotowego produktu.

2. Reprezentacja architektury

Architektura reprezentowana jest przez perspektywę przypadków użycia, perspektywę logiczną oraz omówienie zagadnień dotyczących rozmiary, wydajności i jakości gotowej aplikacji.

3. Cele i ograniczenia architektury

Gra ma być samodzielnym programem działającym na komputerze gracza (nie przewiduje się żadnych dodatkowych systemów).
Gra powinna działać na komputerze klasy PC, wyposażonym w system Windows.

4. Perspektywa przypadków użycia

Perspektywa przypadków użycia przedstawia zbiór funkcjonalności branych pod uwagę przy realizacji projektu.
Wszystkie przypadki użycia jakie projekt powinien realizować znajdują się w Specyfikacji przypadków użycia, w pliku 05_specyfikacja_przypadkow_uzycia.pdf

5. Perspektywa logiczna

Sekcja ta ma za zadanie rozłożenie systemy na podsystemy, pakiety i klasy.

5.1 Streszczenie

Następny podpunkt zawiera opis pakietów systemu, a kolejny – dokładne opisanie klas należących do tych pakietów.

5.2 Pakiety

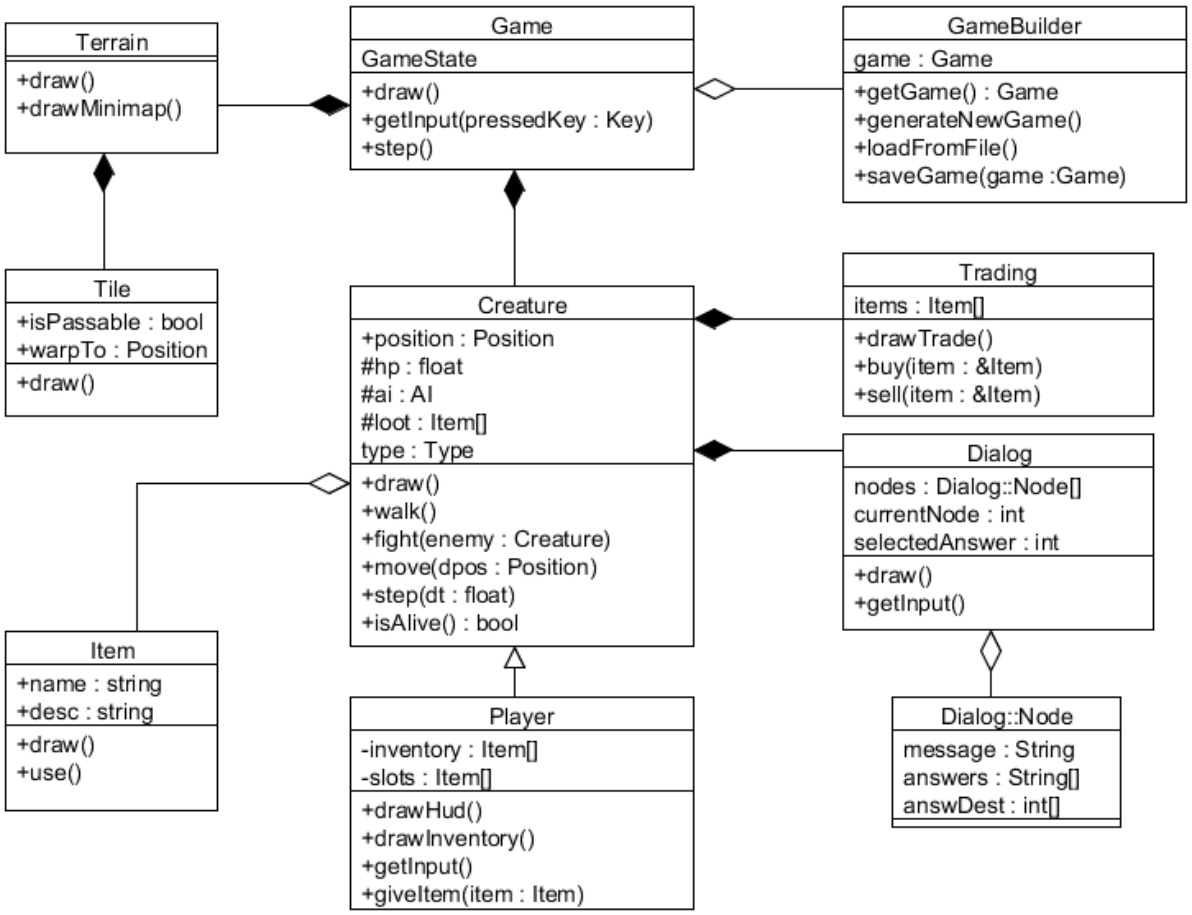
Ze względu na prostotę projektu możemy wyróżnić tylko dwa pakiety: Pakiet Gra i pakiet Postacie.

5.2.1 Opis Pakietów

Gra	
Opis:	Główny pakiet projektu wszystkie akcje użytkownika są przetwarzane przez ten pakiet.
Powiązane klasy:	Game, Tile, Terrain,
Relacje:	Główny pakiet projektu, zależny od pakietu Postacie
Podpakiety:	Brak

Postacie	
Opis:	Pakiet zawiera klasy reprezentujące postacie występujące w grze: postać gracza, postacie niezależne oraz przeciwników
Powiązane klasy:	Player, NPC, Enemy
Relacje:	Zależny pakietowi Gra
Podpakiety:	Brak

5.3 Opis klas występujących w projekcie



Własność	**Opis**
Nazwa	Game
Opis	Podstawowa klasa, spajająca pozostałe klasy
Odpowiedzialności	Zarządza innymi obiektami
Relacje	Korzysta z klas: Terrain, Player, NPC, Enemy
Metody	Draw() : rysuje okno gry getInput(pressedKey : Key) : pobiera wejście z klawiatury

Atrybuty	GameState : reprezntuje stan gry (INGAME – eklsploracja terenu, INVENTORY – przegląd ekwipunku, ATRIBUTES - przeglad statystyk ,DIALOG – rozmowa gracza z NPC, EXIT – wyjscie z gry , GAMEMENU – menu w grze)
Wymagania specjalne	Brak

Własność	**Opis**
Nazwa	GameBuilder
Opis	Generator gry.
Odpowiedzialności	Tworzy świat gry, zapisuje i doczytuje go z i do pliku
Relacje	Brak
Metody	getGame() : zwraca utworzony obiekt Game generateNewGame() : generuje nową grę loadFromFile() : ładuje stan gry z pliku saveGame() : zapisuje bieżący stan gry do pliku
Atrybuty	Brak
Wymagania specjalne	Brak

Własność	**Opis**
Nazwa	Tile
Opis	Reprezentuje kafelek.
Odpowiedzialności	Pozwala na rysowanie podstwawowych elementów mapy, determinuje poruszanie postaci w grze.
Relacje	Używana przez klasę Terrain
Metody	Draw() : rysuje kafelek.
Atrybuty	IsPassable : pozwala stwierdzić czy postać może przejść przez dany kafelek, w : współrzędne miejsca do którego prowadzi kafelek (np. w przypadku schodów, drzwi, itp.)
Wymagania specjalne	Brak

Własność	**Opis**
Nazwa	Terrain
Opis	Reprezentuje mapę świata gry.
Odpowiedzialności	Pozwala na rysowanie mapy gry, determinuje poruszanie postaci w grze.
Relacje	Zależna od klasy Tile, używana przez klasę Game
Metody	Draw() : rysuje całą mapę DrawMinimap() : rysuje minimapę
Atrybuty	Brak
Wymagania specjalne	Brak

Własność	**Opis**
Nazwa	Creature
Opis	Reprezentuje postać – np. NPC, przeciwnika
Odpowiedzialności	Wyświetlanie i poruszanie się postaci
Relacje	Klasa bazowa dla Player
Metody	Draw() : rysuje postać, move(dpos : Position) : porusza postacią fight() : walczy z danym wrogiem walk() : stara się poruszyć postacią w danym kierunku i reaguje w odpowiedni sposób step(dt : float) : wykonuje krok symulacji isAlive() : czy dana postać wciąż żyje
Atrybuty	Position : położenie postaci, HP : ilość pozostałych punktów zdrowia
Wymagania specjalne	Brak

Własność	**Opis**
Nazwa	Player
Opis	Reprezentuje postać gracza
Odpowiedzialności	Przechowywanie informacji o statystykach gracza
Relacje	Dziedziczy po Creature, używana przez Game
Metody	DrawHud() : rysuje HUD drawInventory() : rysuje ekwipunek gracza getInput() : pobiera wejście z klawiatury giveItem(item : Item) : daje postaci przedmiot
Atrybuty	Inventory : przechowuje przedmioty danej postaci slots : aktualnie używane przedmioty
Wymagania specjalne	Brak

Własność	**Opis**
Nazwa	Item
Opis	Reprezentuje obiekt (przedmiot)
Odpowiedzialności	
Relacje	Obiekt posiadany przez klasy dziedziczące po Creature
Metody	Draw() : rysuje obiekt na ekranie
Atrybuty	Name : nazwa przedmiotu Desc : opis przedmiotu
Wymagania specjalne	Brak

Własność	**Opis**

Nazwa	Dialog
Opis	Reprezentuje dialog z NPC
Odpowiedzialności	Obsługa rozmowy gracza z NPC, przechowywanie drzewa dialogów
Relacje	Obiekt posiadany przez NPC
Metody	Draw() : wyświetla wypowiedź NPC i opcje dialogowe (odpowiedzi) getInput() : pobiera wejście z klawiatury
Atrybuty	Brak
Wymagania specjalne	Brak

Nazwa	Menu
Opis	Reprezentuje Menu zarówno główne jaki i menu w grze
Odpowiedzialności	Obsługa rozpoczacia gry/wyjscia z gry /wznowienia/zapisu/wczytywania/opcji.
Relacje	Obiekt posiadany przez Game
Metody	Draw() rysuje menu drawMenuGame() rysuje menu w grze setOutsideGame() ustawia zmienna aby gra mogla się rozpoczac isOutsideGame() sprawdzenie czy gra dziala Action() obsluga klawiatury w menu glownym ActiongameMenu() obsluga klawiatury w menu gry shouldExit() funkcja odpowiadajaca za zamkniecie gry
Atrybuty	Brak
Wymagania specjalne	Brak

6. Opis interfejsu

Opis interfejsu znajduje się w Projekcie interfejsu, w pliku 08_projekt_interfejsu.pdf .

7. Rozmiar i wydajność

Projekt składa się z pojedynczego pliku wykonywalnego działającego na komputerze gracza. Ze względu na prostotę projektu przewiduje się niewielkie wymagania sprzętowe.

8. Jakość

Architektura systemy spełnia wymagania odnośnie jakości, jak opisano w Specyfikacji wymagań projektu i Specyfikacji dodatkowej.