Imię i nazwisko: Paweł Tymoczko

Grupa (dzień, godzina): Środa 8:00

1. Temat aplikacji (*np. gra detektywistyczna, symulator lotu na Marsa):*
Gra typu rogue-like, inspirowana grą Darkest-Dungeon.

2. Czynności wykonywane przez użytkownika pod względem MERYTORYCZNYM (*np. podejmowanie decyzji i rozwiązywanie zagadek, określanie parametrów lotu*):
Użytkownik będzie mógł stworzyć postać, o określonych punktach zdrowia, ekwipunku i zdolnościach. Każda rozgrywka rozpocznie się wybraniem lokacji, do której wyrusza główny bohater. Zadaniem użytkownika jest poruszać się po wygenerowanej mapie swoją postacią oraz wchodzić w interakcję z obiektami umieszczonymi na planszy (zbierać przedmioty ekwipunku, eliksiry) tak, aby znaleźć wyjście. W momencie napotkania potworów na mapie, użytkownik zmuszony jest do podjęcia walki, wykorzystując swoje wyposażenie. Bitwy odbywają się turowo (raz ruch wykonuje przeciwnik, a raz główna postać użytkownika). Jedna tura składa się z użycia przedmiotu, albo rzucenia zaklęcia, albo wykonania podstawowego ataku. W momencie znalezienia wyjścia, użytkownik pomyślnie zalicza odwiedzoną lokalizację. Po zakończeniu misji, użytkownik może dostosować swój ekwipunek, poprzez zakupy w sklepie, oraz wyleczyć się poprzez odwiedzenie szpitala. Po powtórnym przygotowaniu, użytkownik może znów wysłać swoją postać na wyprawę.

3. Czynności wykonywane przez użytkownika pod względem TECHNICZNYM (*np. klikanie przycisków na klawiaturze, dostarczanie plików z parametrami wejściowymi)*:
Zarówno poruszanie po interfejsie menu jak i interfejsie gry, będzie się odbywać tylko i wyłącznie z wykorzystaniem klawiatury. Instrukcje dla gracza będą przekazywane poprzez okno konsoli.

4. Oczekiwane rezultaty (*np. gra kończy się prawidłowym lub nieprawidłowym wskazaniem mordercy; symulacja zwraca informację o tym, czy udało się dolecieć na Marsa)*:
Gra kończy się w momencie odwiedzenia przez postać użytkownika wszystkich lokacji.

5. Proszę zidentyfikować jeden przypadek wykorzystania polimorficznej metody, który na pewno trzeba będzie umieścić w programie *(np. Metoda [nazwa] w klasie/interfejsie [nazwa] i klasach dziedziczących – w zależności od klasy, metoda ta będzie...)*:

Interfejs „ruch tury” będzie udostępniał metodę „wykonaj ruch”. Klasy dziedziczące „podstawowy atak”, „atak zaklęciem”, „wypij eliksir” oraz „pomiń” będą implementować konkretne funkcjonalności. Tak, więc użycie metody z klasy „podstawowy atak” będzie jedynie zadawało obrażenia wrogowi, natomiast użycie funkcji z klas „atak zaklęciem” oraz „wypij eliksir” będzie wymagało wybrania konkretnego zaklęcia/eliksiru. W przypadku użycia eliksiru, konieczne będzie również sprawdzenie czy takowy jest dostępny w ekwipunku postaci.

6. Proszę zidentyfikować jakiś rodzaj relacji między klasami inny niż dziedziczenie wykorzystywane w punkcie 5, który trzeba będzie umieścić w programie *(np. Klasa [nazwa1] połączona z klasą [nazwa2] przy pomocy dziedziczenia/agregacji/kompozycji/zależności...)*:
Agregacja, element wyposażenia a ekwipunek: Każdy *element wyposażenia* (klasa miecz/różdżka, klasa zbroja/szata, klasa rękawice itp.) będzie przechowywany przez klasę *„ekwipunek”*.
Kompozycja, postać a ekwipunek: Klasa postać będzie składała się z ekwipunku.

7. Proszę zidentyfikować i nazwać trzy klasy, które nie zostały wymienione w punktach 5 i 6, a które na pewno trzeba będzie umieścić w programie *(np. Klasy [nazwa1], [nazwa2], [nazwa3])*:
Interfejs *„potwór”* oraz jego implementacje – opisuje statystyki przeciwników, przechowuje „interfejs tury” (używany ruch do walki).
Klasa *„mapa lokacji”* – przechowuje wymiary danej lokacji oraz rozmieszczenie poszczególnych przedmiotów, pokoi z potworami.
Interfejs „*lokacja*” – przechowuje stan o tym czy lokacja została odwiedzona, generuje mapę, udostępnia metodę *„odwiedź”*, pozwalającą na rozpoczęcie wyprawy.

8. Miejsce na ewentualne uwagi lub pytania do prowadzącego: