**Софийски университет „Св. Климент Охридски“  
Факултет по Математика и Информатика**

**Проект**

по

Функционално програмиране

УЧЕБНА ГОДИНА 2022/2023

**ТЕМА**

Текстова приключенска игра

**Изготвил: Радослав Каратанев**

**Факултетен номер: 3MI0800036**

**Група: 4**

**Съдържание:**

1. **Общо описание на играта**
2. **Помощни функции**
3. **Структура на играта**
4. **Битки**

**Общо описание на играта**

Играта започва с командата “game start”. Играчът се намира в своя лагер, като вижда около себе си всички места, до които може да се придвижи. Пример:

You can go to:

1. Forest
2. Mines

Ако играчът желае да се придвижи до 1), е нужно да впише Forest, а ако желае да се придвижи до 2) – Mines. Играчът също има и предмети, които може да намира и взима от повечето местоположения. Предметите представляват оръжия с произволна сила. При пътуване от едно до друго място, играчът има възможност да бъде атакуван. Той може да избере да бяга от нападателя си, както и да се изправи срещу него.

**Помощни функции**

За създаването на играта са реализирани следните помощни функции:

1. removeOne/removeAll – премахват предмети от даден списък
2. removeFromLocation – премахва всички предмети от дадено местоположение
3. getItems – връща списък с всички предмети, намиращи се на даденото местоположение
4. enumerate (enumerateWeapons) – при подаден списък от тип String(Weapon), го номерира от 1
5. accessible – при подадено местоположение, връща към кои други местоположения има директна връзка
6. getStatsOfAWeapon – При подадени име на предмет и списък от предмети, връща силата на предмета с това име
7. getRandomElement – връща случаен елемент от даден списък

**Структура на играта**

Не от всяко място, където се намира играча, той може да се придвижи до всяко друго. Връзките от местоположение до местоположение се дефинират в \_map (Ln 34). \_map представлява списък от наредени двойки, като всяка двойка показва от кое до кое местоположение има директен път. На всяко място в света на играта има възможност да има предмети, които играчът може да вземе. Това се осъществява с take. Всички предмети са от тип Weapon, който представлява наредена двойка от тип String (име на предмета) и Int (силата на предмета). Картата на играта с всички места и предмети, намиращи се в тях, се дефинира в \_contents (Ln 16), като представлява списък от двойки. Първия елемент на всяка двойка е името на местоположението, а вторият елемент на двойката е списък от всички предмети, намиращи се в него. Всички местоположения и предмети, използвани в играта също трябва и да се добавят съответно в locations (Ln 31) и \_weapons (Ln 28). Играта се започва с командата “game start”, като функцията game имитира цикъл, който приема различни стойности в параметрите си при всяко решение на играча. Start задава началните стойности на тези параметри. По всяко време може да се напише елемент от exitWords (Ln 170) за да се излезе от играта (Пример: q).

Можете да видите и <https://github.com/radkotoktv/Haskell-Project>. Репо-то за момента е Private, за избягване на плагиатство.

**Битки**

Битките с противници се реализират с функцията fight:

1. Описание на противниците – всеки противник е от тип Enemy. Този тип съдържа 3 основни елемента:
   1. Race – расата на нападателя ни
   2. Health – колко живот има
   3. [Attack] – списък от силата на различните атаки, които може да използва
2. Описание на играча – от тип Player, който съдържа:
   1. Health – животът на играча
   2. [Weapon] – списък от всички оръжия, които притежава
3. Визуализация на битката – функцията fight използва помощните функции printEnemy и printPlayer за да визуализира всички данни за участниците в битката. След това, чака играча да избере с кой предмет ще атакува. Пример: Weapons:  
    1. Wooden sword (5 damage)  
    2. Shield (2 damage)  
   За да атакува с 1), играчът трябва да въведе Wooden sword, а за да атакува с 2) – Shield. Животът на противника пада, в зависимост от това с какво оръжие е бил нападнат. Едновременно с това, срещуположната страна избира случайна атака от списъка си от 1c. и животът на играча пада с толкова, колкото е била силата на тази атака. Тези стъпки се повтарят, докато животът на единия в двубоя не падне под или равен на 0, при което се решава победителя.

За тестова битка, може да се извика fight (testEnemy, testPlayer).