



**СУ “Св. Климент Охридски”,  
ФМИ – Софтуерно инженерство  
Курсов проект по Обектно-ориентирано  
програмиране**

# **Console Game**

Даниел Кирилов Иванов, Факултетен № 61796

## Съдържание

1. Въведение
2. Описание на приложените алгоритми
3. Описание на програмния код
4. Използвани технологии

## Въведение

Целта на проекта е да бъде създадена конзолна игра, подобна на TBS (Turn-based strategy) игрите. В играта ще се изправят една срещу друга две армии, като битките се развиват на ходове. Видовете единици са: Peasant, Footman, Archer, Griffon и Hero. Игралното поле се представя чрез матрица 10x10, като нашата и противниковата армията се разполагат върху нея. Героят на играча и противниковият герой са извън бойното поле. Единиците имат различна stamina - колко квадратчета далеч могат да се придвижат, в радиус от текущата позиция:

Peasant – 3, Footman – 4, Archer – 3, Griffon – 5, Hero не може да се мести по полето.

Единиците имат различен обхват на атака, в радиус от текущата им позиция: Peasant – 1, Footman – 1, Archer – може да напада всяка противникова единица по полето, независимо от разстоянието, Griffon – 2, Hero – всяка противникова единица.

Освен, че играчът губи умрелите по време на игра единици, ако след края на дадена битка е загубил играта - губи и 100g. Ако е спечелил - печели 200g. Играта приключва когато един от играчите остане без единици на полето.

# Описание на програмния код

**Кодът е разделен в няколко пакета :**

- medieval-street-fight.engine
- medieval-street-fight.units
- medieval-street-fight.main от който се стартира играта

**1.Пакетът medieval-street-fight.engine :**

- engine.BattleField.java
- engine.DijkstraAlgorithm.java
- engine.Edge.java
- engine.Game.java
- engine.Graph.java
- engine.Player.java
- engine.Units.java
- engine.Vertex.java

**BattleField.java :**

Генерира матрица 10x10, която се използва за бойно поле. В него са имплементирани функциите:

- printField() (принтира матрицата по атрактивен начин в Game-Field.txt)
- generateGraph(), generateEdges(), generateOccupiedVertices(), generateVertices() (тези 4 функции осигуряват генерирането на граф, представящ бойното поле. Този граф се използва от алгоритъма на Дейкстра, който използвам за да намеря най-кратките пътища от една единица до друга)

**DijkstraAlgorithm.java** - При подаден граф, намира най-кратките пътища от точка до точка.

**Edge.java:** При нова инстанция на класа се създава ребро, което пази информация за върховете, които свързва.

**Game.java:** Класът Game.java имплементира:

- менюто за играта.
- менюто за магазина и купуване на единици.
- ходовете, които изпълнява базовия изкуствен интелект.
- четенето на командите зададени от играча (преместване и атакуване)
- край на играта.

Graph.java: Притежава списък от върхове и ребра.

Player.java: При инстанцирането се създава играч, който пази информация за:

- Единиците от всеки вид, които притежава.
- Дали тези единици са използвани вече в текущия ход.
- Дали играча е на ход.

Units.java: енумерация, в която се съдържат всички видове единици.

Vertex.java: При нова инстанция на класа се създава нов връх, който пази информация за координатите си и каква единица седи на тези координати.

2. Пакетът medieval-street-fight.units :

- Archer.java
- Creature.java
- Footman.java
- Griffon.java
- Hero.java
- Peasant.java

При нова инстанция на класовете Archer, Peasant, Footman или Griffon се създава нова единица с:

- щета
- защита
- здраве
- мана
- издръжливост
- шанс за критичен удар
- максимална атакуваща дистанция

## Технологии

Проектът е написан на езика Java

Използвал съм :

Eclipse IDE for Java Developer

Version: Luna Service Release 2 (4.4.2)

(c) Copyright Eclipse contributors and others 2000, 2015. All rights reserved