

# Курсов проект

---

По

Обектно-ориентирано програмиране II-ра част

**Изготвил:** Петър Петков

**Факултетен №:** 18621811

**Специалност:** СИТ

**Курс:** Втори

**Група:** 6

**Половина:** А

# I. Задание

## Проект 1

### Diablo 0.5

(изготвил: Александър Шумаков)

Една от най-популярните RPG поредици се казва *Diablo*. Третото допълнение бе дългоочаквано и придоби голям успех след пускането на първото разширение: *Reaper of Souls*. Но зад успеха на тази вече култова класика стои дълга история на разработка – първата част е пусната на пазара на 31.12.1996г. По онова време компютърните игри не са били с толкова полирани визуални и звукови качества и ефекти, както са днес. Били са прости – базирали са се на една основна идея и са се развивали около нея. Някои от игрите, смятани за най-велики и въздействащи и до днес, всъщност не представляват нищо повече от един конзолен интерфейс.

Вашата задача е да направите примитивна версия на играта *Diablo*.

За целта ще ви е нужно да реализирате следните класове и интерфейси:

#### Character

**Character (интерфейс):** този интерфейс ще представлява основните данни и функционалности за всеки един герой;

**Данни:**

- **Име (name):** названието на един бъдещ легендарен войн;
- **Ниво (level):** кое ниво е героят ни в момента;
- **Жизнени точки (HP):** оставащият живот в героя;
- **Сила (strength):** колко физическа сила притежава;
- **Интелект (intelligence):** магическата способност на даденият герой;

**Функции:**

- **Нападение (attack):** нанасяне на щета върху враг;
- **Отбрана (defend):** получаване на щети, като от точките за атака на противника се изкарват толкова проценти, колкото е **strength** показателя на героя;
- **Порастване (levelUp):** функция, която се извиква автоматично при убиване на  $2^{<level>}$  противника; подсилва показателите спрямо пропорционалното им разпределение при създаването на героя. За тази цел се разпределят общо 5т. на 2 показателя. (Забележка: жизнените точки се покачват автоматично с 10% от текущата им стойност)

Например:

При начална статистика от сорта на 10 **strength** и 4 **intelligence**, то ще имаме следното пресмятане:

$\text{<нова стойност на показател>} ::= [(\text{<първоначална стойност>} / 14) * 5] + \text{<текуща стойност>}$

Героите могат да бъдат следните няколко вида:

- **Варварин (Barbarian):** физически могъщ и свиреп, майстор на близкия бой и контактните оръжия. Притежава специална енергия, наречена **ярост (rage)**. Яростта се генерира когато варваринът удари противник (+2т.) или когато противникът уцели варваринът (+3т.). Капацитетът на герой от този вид е до 100т. ярост.

**Начални стойности:**

100HP, 10 strength, 3 intelligence, 0 rage

**Варваринът** се бие по следния начин: изразходва всичката ярост, която притежава, в началото на всяка битка, като яростта му дава бонус (ярост / 5) процента атака за цялата битка с даденият противник. Базовата щета, която нанася варваринът, е  $\text{<точки сила>} + 0.2 * \text{<точки интелект>}$ . Натрупаните по време на конкретна битка точки ярост се използват за следващата битка.

- **Магьосник (Sorcerer):** мъдър, спокоен и непоклатим психически, почитател на древните тъмни учения. Притежава специална енергия, наречена мана (**mana**). Маната се възвръща на максимум след всяка битка. Максималният капацитет мана, който един магьосник може да има, се увеличава с 10% при всяко порастване (подобно на жизнените точки).

**Начални стойности:**

70HP, 4 strength, 12 intelligence, 100/100 (100 точки в началото от 100 точки начален капацитет) mana

**Магьосникът** се бие по следния начин: при всяка атака нанася щети, определени от следната формула:  $[50\% + (\text{<текуща мана>} / \text{<капацитет мана>}) * \frac{3}{4}] * \text{<точки интелект>}$ . След нападение, точките мана на разположение на магьосника спадат с 10% от капацитета.

- **Ловец на глави (bounty hunter):** изобретателен и ловък, експлоатиращ и най-незабележимите пукнатини в защитата на врага. Не притежава специална енергия, но пък разполага с ловкост (**agility**). Точките ловкост се повишават с 4 на всяко порастване.

**Начални стойности:**

80HP, 9 strength, 6 intelligence, 10 agility

**Ловецът на глави** се бие по следния начин: всяка трета атака, която нанася в дадена битка, нанася на врага  $\text{<точки ловкост>} + 0.4 * \text{<точки интелект>} + 0.6 * \text{<точки сила>}$  точки щета. Останалите 2 атаки нанасят  $0.8 * \text{<точки ловкост>}$  точки щета.

Enemy

**Enemy (интерфейс):** този интерфейс ще предоставя основните данни и функционалности за всеки един противник;



## Данни:

- **Име (name):** названието на един свиреп звяр
- **Жизнени точки (HP):** оставащият живот във врага
- **Сила (strength):** колко физическа сила притежава
- **Интелект (intelligence):** магическата способност на даденото чудовище

## Функции:

- **Нападение (attack):** нанасяне на щета върху герой
- **Отбрана (defend):** получаване на щети, като от точките за атака на противника се изкарват толкова проценти, колкото е strength показателя на чудовището.

Враговете могат да бъдат следните няколко вида:

- **Скелет (Skeleton):** малоумно чудовище, задвижвано от неznайна сила; не представлява заплаха за който и да е истински герой

### Начални стойности:

32HP, 3 strength, 0 intelligence

Скелетът се бие по следния начин: нанася щети колкото е силата му + 1/10 от интелекта на врага

- **Еретик (Heretic):** изгубена душа в тялото на човек, подмамен от пътя на черната магия; залъгва своите опоненти и ги напада емоционално

### Начални стойности:

20HP, 1 strength, 4 intelligence

Еретикът се бие по следния начин: нанася щети колкото са точките на най-ниският показател на опонента му + точките интелект на самият еретик.

- **Макромант (Necromancer):** полужив труп със задълбочени познания в забранените изкуства, играещ си с живота и смъртта; може да призовава поданик скелет, който да го отбранява, докато той се готви да нанася удари

### Начални стойности:

65HP, 2 strength, 7 intelligence

Макромантът се бие по следния начин: на първият рунд от битката (а след това и на всеки три рунда) той призовава нов свой поданик скелет, който да поема ударите вместо него. Нанася щети колкото са точките HP на скелета-поданик / 10 (ако има такъв, ако няма – 0) + своите точки интелект.

- **Демон (Diablo):** първичното изчадие, породило всичко зло на планетата; породено от ерес и смърт (т.е. да се имплементира като наследник на класовете Еретик и Скелет)

### Начални стойности:

350HP, 8 strength, 4 intelligence

Демонът се бие по следният начин: редува по една атака като на скелет и като на еретик.

Да се имплементира интерактивен режим на играта, който включва:

- Избор на герой
- Карта (двумерен масив), по която са разположени разнообразни противници;
  - когато се опитаме да стъпим върху поле на противник, то героят го атакува и след това, ако противникът е все още сред живите, той отвръща с атака;
  - ако се опитаме да стъпим на невалидно поле - да се връща пояснително съобщение защо това е "невалиден ход" и да се дава възможност за повторно въвеждане.
- Стандартен изход на актуална информация за картата и последиците от последния ход на играча

## II. Кратко описание на програмата

### Предназначение

Програмата представлява RPG игра, която предоставя интерфейс чрез конзола, която приема една след друга команди, които да предприеме. След избиране на герой и генериране на (произволно разположено) игрално поле, играчът има възможност да обикаля по поле в стил шахматна дъска, с размери 10x10 плочки. По полето има разположени на случаен принцип 20 противника, по 5 от всеки тип, описан по-горе. Целта на играча е да победи всички противници, без да бъде победен на свой ред.

Програмата предоставя принтиране на информация за полето, играча и справки за проведена битка (combat log), които се принтира докато се провежда самата битка.

### Данни, с които работи програмата

Програмата изисква да се въвеждат команди в конзолата, отговарящи на инструкциите, които излизат в нея. При нормално изпълнение, първо се изисква да се избере класа на героя (чрез числа, които са описани в конзолата). След това се иска име на героя, което може да бъде каквото поискаме. След това се въвеждат предефинирани команди, които могат да бъдат изписани чрез командата "help".

### Възможни грешки и защита, включена в програмата

Програмата може да работи с множество формати на входните данни, включително и символи, които не са в ASCII таблицата. Командите се разпознават автоматично и се отхвърлят, ако не са разпознати или невалидни.

Недостатък на програмата е генерацията на игрално поле. Понеже в C++ функцията, която генерира случайно число не функционира правилно, винаги полето ще се създава с еднакво разположени врагове. Затова се използва малък заобиколен начин, който опитва отново с нови числа, докато не намери число, което не е еднакво на предишното. В резултат на това генерацията на поле често е бавна, като отнема от 30 секунди до 1 минута. Този проблем не фигурира в C# версията на проекта, предоставена в архива.

### Възможности за развитие на програмата

Програмата може да бъде развита по много различни начини, които включват:

- Добавяне на повече класове на герой
- Добавяне на повече типове на врагове
- Създаване на поле, базирано на входни параметри като размер, брой на врагове и други.
- Добавяне на повече видове интеракция между врагове и герой (париране, избягване на удар и подобни)
- Добавяне на PVP (битки на 2 героя един срещу друг)

## III. Обща структура на програмата

Програмата е разделена в няколко класа обекти, които следват йерархична структура.

### • Клас **Entity**

- Представява главния клас на всички същества в програмата.
- Отговаря за базовата имплементация на действия като атакуване, защитаване и инициализиране на параметри на съществото
- **Функции:**
  - **set\*(\*) / get\*():** сетъри и гетъри за property-та на класа
  - **Attack(Entity& entity):** функция, която изпълнява базовата атака на същество срещу друго
  - **Defend(double damageToInflict):** функция, която намаля подадената като параметър щета базирано на силата на съществото.
  - **StartCombat() & EndCombat():** функции, които управляват състоянието на героя (в битка, или в покой)
  - **print():** Функция, която имитира ToString() принтиране на съществото с всички негови статистики.

### • Клас **Character**

- Представява главния клас за трите вида герои. Наследява **Entity**
- Отговаря за нивото на героя и покачването на това ниво, а също и за следенето на брой убити същества (нужно за вдигане на ниво)
- **Функции:**
  - **LevelUp():** Функцията, която изпълнява базовото действие на вдигане на ниво. Приема параметри, които са началните статистики на героя. Нужни са за да се изчислят новите

статистики след вдигане на ниво, но не се знаят, защото класа не знае какъв тип е героя и какви са началните му статистики.

- Клас **Barbarian**
  - Представлява първия от класове герои. Наследява **Character**
  - Имплементира действия, специфични за този герой (Rage) и добавки към базови функционалности на атакуване и защитаване.
- Клас **Sorcerer**
  - Представлява втория от класове герои. Наследява **Character**
  - Имплементира действия, специфични за този герой (Mana) и добавки към базови функционалности на атакуване и защитаване (намаляване на маната и пълненето и след края на битката).
- Клас **Bounty Hunter**
  - Представлява третия от класове герои. Наследява **Character**
  - Имплементира добавки към базови функционалности на атакуване и защитаване (броене на поредицата от нанесени удари, всеки трети има специално изчисление на щетата).
- Клас **Enemy**
  - Представлява главния клас за четирите вида врагове. Наследява **Entity**
- Клас **Skeleton**
  - Представлява първия от тип врагове. Наследява **Enemy**
  - Имплементира добавки към базови функционалности на атакуване и защитаване.
- Клас **Heretic**
  - Представлява втория от тип врагове. Наследява **Enemy**
  - Имплементира добавки към базови функционалности на атакуване и защитаване.
- Клас **Necromancer**
  - Представлява третия от тип врагове. Наследява **Enemy**
  - Имплементира добавки към базови функционалности на атакуване и защитаване (броене на поредицата от нанесени удари, функционалността с поддържане на собствена армия от скелети, описана в спецификацията).
- Клас **Demon**
  - Представлява четвъртия от тип врагове. Наследява **Enemy**
  - Имплементира добавки към базови функционалности на атакуване и защитаване (редуване на атаки като скелет и като херетик).
  -
- Клас **Tile**
  - Представлява Най-малката единица в игралното поле. Плочка, на която могат да стоят герой и враг.

- Клас **Field**
  - Представява игралното поле, матрица от обекти от тип **Tile**
- Клас **GameState**
  - Представява главното състояние на играта, поддържа цикъла, който управлява самата игра, генерацията на полето, поставянето на враговете и играча, битките между същества.

## IV. Сорс код

Намира се в папка `Diablo_C` в предоставения архив. Сорс кода на програмата, написана на C# оригинално, се намира в папка `Diablo_CS`.