

# Heroes of Might and Magic alá Bio

*Jakub Suchý*

## Programátorská dokumentace

### Úvod

HoMaM alá Bio je 2D hra kopírující v některých věcech HoMaM I. Hra je napsaná v C# za pomoci knihoven .NET a XNA. Jedná se o jednookenní aplikaci, která má předem danou velikost a nelze ji změnit.

### Přesné zadání

Ve výsledku to bude tak nějak lehce osekana znama hra Heroes of Might and Magic.

Schopnosti:

- 2D
- Ovládání myši
- SinglePlayer
- Uložení/Nahrání hry

Algoritmy:

- Vyhledávání nejkratší cesty při pohybu
- Minimax při souboji s potvorami

Technologie:

- C#
- XNA

Platforma: Windows

### Použité algoritmy a technologie

- Nejdůležitější použitý algoritmus je vyhledávání nejkratší cesty v mřížce. Použil jsem již mnohokrát ztvárněný algoritmus A\*, který jsem použil jako zápočet na jiný předmět.

Nebudu sem algoritmus vypisovat, protože přikládám celou dokumentaci toho algoritmu spolu s touto dokumentací.

- Další technologií kterou jsem použil je TileEngine. Ten slouží pro vytváření vrstev mapy, vykreslování mapy a uchování všech informací o mapě.

## Diskuse výběru algoritmu

Pro vyhledávání cesty v grafu(mřížce) je mnoho algoritmů. Dalo se použít cokoliv, např. vlna, backtracking, Dijkstra, atd.. Ale já si vybral A\*, protože už jsem ho měl naprogramovaný jako zápočtový program na ADS I. Přímo za účelem jeho následovného použití v této hře.

## Program

Program je opravdu obsáhlý. Jeho základními stavebními prvky je pár hlavních komponent.

Session - je to třída, která se dá instancovat pouze jednou a to na začátku programu a obsahuje všechny důležité statické prvky. Čili slouží ke globalizaci veškerých potřebných dat.

ScreenManager - tato třída se stará o vykreslování, změnu a zásobník screenů. Díky této hře se dá ve hře pohybovat meníčkama dopředu a např. klávesou Esc zpět.

TileEngine - je to soubor tříd, které se starají o uchování všech vrstev mapy, její vykreslení a manipulaci s ní, co se týče obsluhy jednotlivých políček.

Controls - je to soubor tříd, které akorát kopírují některé formulářové prvky. Jako je například: Label, PictureBox, Button, atd...

Serialization - je to soubor tříd, které pomáhají ukládat a načítat hru z vnějších souborů. Díky tomu jde hru uložit a nahrát.

Další části už zde nebudu vypisovat, protože kód je až přehnaně dopodrobna okomentovaný.

## Průběh prací a programová řešení

Ze začátku jsem čerpal z jednoho tutoriálu pro tvorbu RPG her, který uvedu v referencích. Takže ze začátku bylo vše přehledné a všechny objekty byli krásně zabalené a nepřístupné. Ovšem, když jsem vyčerpal vše co se dalo a začal řešit problémy adaptované na mou hru, tak jsem pomalu zjišťoval, že skoro všechno musím otevřít. Hlavně kvůli serializaci a deserializaci. Proto je kód co se týče objektů skoro plně přístupný odevšad, což v objektovém programování není dobře a mohlo to být napsáno lépe.

Co se týče průběhu práce. Tak díky výše řečeného tutorialu jsem měl ze začátku práci snadnou. Ovšem ke konci už jsem si občas nevěděl rady, jak vůbec některé věci vyřešit. Nakonec jsem je ale vyřešil. A hra je tak dotažena do konce.

## Co nebylo doděláno

Bohužel jsem nesplnil ze zadání boj s potvorama řešený za pomoci Minimaxu, kvůli složitosti vykreslování scény a nedostatku času k naprogramování Minimaxu. Dále se hra vůbec nepodobá HoMaMu I skoro vůbec. Vypustil jsem kompletně suroviny, hrady. Hru více hráčů. Osekal jsem co se dalo, kvůli komplexnosti a velikosti programu.

## Závěr

Musím říci, že volba této hry byla velice špatná. Protože vytvořit kopii HoMaM I není tak snadné, jak jsem si představoval. Nakonec jsem po dohodě hru dostatečně osekal a dohnal do konce. Naučil jsem se mnoho nových věcí, třeba jak funguje takový TileEngine, Serializace, co jsou to delegáti a eventy, a naučil jsem se trošku s Frameworkem XNA. Práce mě bavila, jen škoda, že jsem si projekt časově rozvrhnul špatně, proto je vše tak osekáné. Od tohoto projektu bych rád odsunul ruce a přešel do další dimenze na 3D.