



FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO

KATALÓG POŽIADAVIEK

vizualizácia algoritmu **Minimax**

Stanislav Krajčovič Martin Maco Miloš Polakovič Vlastimil Starec
zimný semester 2015/2016

Obsah

1	Úvod	2
1.1	Účel tohto dokumentu	2
1.2	Rozsah produktu / aplikácie	2
1.3	Slovník pojmov, skratky	2
1.4	Referencie	2
1.5	Sumár zvyšku dokumentu	2
2	Všeobecný opis	3
2.1	Perspektíva produktu	3
2.2	Funkcie produktu	3
2.3	Užívateľské charakteristiky	3
2.4	Všeobecné obmedzenia	3
2.5	Predpoklady a závislosti	3
3	Špecifické (funkčné) požiadavky	4
3.1	Dostupnosť	4
3.2	Použitelnosť	4
3.3	Korektnosť	4
3.4	Efektivita	4

1 Úvod

1.1 Účel tohto dokumentu

Tento dokument slúži ako východiskový bod pre vývoj aplikácie *minimax.py*. Špecifikuje konkrétne požiadavky zadávateľa na objednanú aplikáciu, rovnako ako kľúčové body potrebné pre proces vývoja. Môže byť použitý pri priebežnom rovnako ako aj finálnom posudzovaní zhody vyvíjanej aplikácie so špecifikáciou zadávateľa, no nemá právny význam (nejedná sa o záväznú zmluvu medzi zadávateľom a vývojárskym tímom).

1.2 Rozsah produktu / aplikácie

Výsledným produktom projektu bude GUI aplikácia pre PC, ktorá bude slúžiť na vizualizáciu a interakciu s algoritmom pre umelú inteligenciu známym pod názvom *Minimax* pre účely vzdelávania.

1.3 Slovník pojmov, skratky

<i>Minimax</i> , <i>MinMax</i> alebo <i>MM</i>	Algoritmus používaný najmä pri strategických hrách dvoch hráčov
<i>minimax.py</i>	Pracovný názov vyvíjanej aplikácie. S veľkou pravdepodobnosťou bude mať výsledný softvér odlišný názov.
<i>GUI</i>	<i>Graphical user interface</i> ; grafické užívateľské rozhranie.
<i>PC</i>	Osobný počítač (desktop alebo laptop)
<i>farebná schéma</i>	Kombinácia viacerých farieb použitých na vytvorenie grafického výstupu
<i>unit test</i>	Druh automatizovaného softvérového testu, ktorý testuje <i>units</i> – najmenšie testovateľné jednotky zdrojového kódu (v <i>OOP</i> sú to najčastejšie buď celé rozhrania tried alebo ich jednotlivé funkcie, resp. <i>metódy</i>)

1.4 Referencie

Oboznámiť sa s algoritmom *Minimax* v anglickom jazyku je možné na adrese <http://wiki.pe/Minimax>. Dokumentácie zvolených programovacích prostredí (*Python* a *Cython*) sú dostupné na adresách <https://docs.python.org/> a <http://docs.cython.org/>, obe tiež v anglickom jazyku. Licencia *GNU GPLv3* je v anglickom znení voľne dostupná na adrese <http://www.gnu.org/licenses/gpl-3.0.en.html>. Jej preklad do slovenského jazyka je dostupný na adrese <http://www.gpl.sk/v3/>.

1.5 Sumár zvyšku dokumentu

Sekcia Všeobecný opis načrtá perspektívu aplikácie, sumarizuje jej základnú funkcionálnosť, charakterizuje rôzne druhy jej cieľových užívateľov a pojednáva o všeobecných obmedzeniach. Sekcia Špecifické (funkčné) požiadavky analyzuje špecifické funkčné požiadavky kladené na vývojárov aplikácie.

2 Všeobecný opis

2.1 Perspektíva produktu

Určením aplikácie *minimax.py* je byť používaná na vzdelávacie účely. Môže byť použitá ako učebná pomôcka na prednáškach, rovnako ako interaktívny softvérový študijný materiál pre študentov.

2.2 Funkcie produktu

Grafickým výstupom aplikácie bude vždy vizualizácia momentálneho stavu zvolenej hry a interného stavu algoritmu. Medzi stavmi bude možné v chronologickom poradí prechádzať užívateľskými vstupmi. V iných slovách, algoritmus bude v aplikácii možné graficky príjemným spôsobom krovať.

2.3 Užívateľské charakteristiky

Sú dva druhy používateľov, pre ktorých je aplikácia určená:

Rola	Popis
Prednášajúci	Osoba ktorá problematike už rozumie a používa aplikáciu na vysvetlenie fungovania algoritmu svojim študentom.
Študent	Osoba bez predošlej znalosti problematiky, ktorá buď používa grafické rozhranie aplikácie na pozorovanie a skúmanie stratégie algoritmu, alebo študuje aplikáciu samotnú prostredníctvom jej dokumentácie, zdrojového kódu a existujúcich automatizovaných testov.

2.4 Všeobecné obmedzenia

Relevantnými softvérovými a hardvérovými obmedzeniami na beh aplikácie z hľadiska užívateľa budú najmä použitý *operačný systém* a *zobrazovacie zariadenie*. Tieto obmedzenia a k nim zvolené prístupy pri vývoji sú popísané v podkapitolách Dostupnosť a Použiteľnosť sekcie Špecifické (funkčné) požiadavky.

Právne obmedzenia sa aplikujú v zmysle licencie *GNU GPLv3*. V skratke – *minimax.py* je slobodný softvér, a teda je poskytovaný bez obmedzení slobôd spustiť aplikáciu na akýkoľvek účel, študovať ju či ju upravovať, kopírovať alebo vylepšovať. Jedno právne obmedzenie hodné zmienky kladené na distribúciu upravených verzií aplikácie však je, aby tieto tiež boli distribuované ako slobodný softvér.

2.5 Predpoklady a závislosti

N/A

3 Špecifické (funkčné) požiadavky

3.1 Dostupnosť

minimax.py je aplikácia určená pre vzdelávacie/akademické prostredie, a musí preto byť prístupná každému bez rozdielov. Týmto požiadavkám bol v plánovacej fáze prispôsobený výber softvérovej licencie a voľba použitých programátorských prostriedkov. Vo svetle týchto požiadaviek bude výsledná aplikácia distribuovaná pod slobodnou licenciou *GNU GPLv3* vo forme zdrojového kódu a binárnych spustiteľných súborov pre operačný systém Microsoft *Windows*.

3.2 Použiteľnosť

Grafický výstup aplikácie by mal byť čo najviac univerzálny. Pod týmto sa rozumie to, aby bol použiteľný na čo najširšom spektre výstupných zariadení a svetelných podmienok. Konkrétne musí byť grafický výstup okrem plochých obrazoviek prispôsobený aj premietaniu na plátno v tmavom, alebo naopak presvetlenom prostredí. Vzhľadom na tieto požiadavky bude grafické rozhranie aplikácie realizované tak, aby sa responzívne prispôbovalo rozlíšeniu výstupného zariadenia (resp. veľkosti svojho okna), a to aj za behu aplikácie. Rovnako bude obsahovať možnosť za behu meniť konfiguráciu používanej farebnej schémy, a to podľa svetelných podmienok miestnosti, druhu obrazovky, osobného vkusu a/alebo iných preferencií.

3.3 Korektnosť

Vzhľadom na svoje určenie pre vzdelávanie musí byť aplikácia korektná v zmysle správnosti implementácie algoritmu a jeho grafickej reprezentácie, t.j. nikdy nemôže spraviť nesprávny krok alebo zobrazíť nekorektný výstup. Takáto kvalita softvéru sa dá zaručiť iba dôkladným testovaním kódu, a to tiež iba do istej miery. Preto je jedným z hlavných cieľov vývojárov zodpovedných za samotnú implementáciu algoritmu dosiahnuť čo najvyššie pokrytie relevantného kódu unit testami. Okrem toho bude v rôznych fázach vývoja prebiehať funkčné testovanie užívateľských rozhraní externými testerami.

3.4 Efektivita

Aplikácia musí byť efektívna v zmysle časovej a pamäťovej zložitosti. Toto do istej miery súvisí aj s požiadavkami na dostupnosť a použiteľnosť – sub-optimálna efektivita by obmedzovala ako dostupnosť (spodná latka na “najslabší” hardvér by bola o čosi vyššie), tak aj použiteľnosť (dlhší čas odozvy by mohol použiteľnosti uškodiť).

Týmto kritériám bol prispôsobený výber programovacích jazykov. Väčšina aplikácie bude napísaná v silne dynamicky typovanom programovacom jazyku *Python*. Vzhľadom na exponenciálnu časovú zložitosť algoritmu a real-time charakter aplikácie však budú kritické sekcie kódu musieť byť implementované za použitia statických dátových štruktúr a rutín, keďže takýto prístup dokáže byť rádovo rýchlejší než jeho dynamický protipól. Pre tieto sekcie kódu bol zvolený programovací jazyk *Cython*, ktorý zdieľa mnohé z výhod jazykov *C* a *C++*. Takto napísaný kód bude možné elegantne integrovať so zvyškom aplikačného kódu.