

Licencjacki projekt programistyczny 2011

Program do wykonywania obliczeń statystycznych związanych z grą go

Dokumentacja programisty

Wojciech Jedynek

Wrocław, 28 czerwca 2011

Spis treści

1	Wprowadzenie	3
1.1	Cel dokumentacji	3
2	Organizacja projektu	3
2.1	Ogólny opis	3
2.2	Konfiguracja programu	3
2.2.1	Opis formatu pliku CONFIG	3
2.3	Baza danych	3
2.3.1	Tabela go_stat_data	4
2.4	Struktura modułów	4
2.5	Pozostałe pliki	4
3	Kompilacja i testowanie	5
4	Wykorzystane biblioteki i narzędzia	5
4.1	Biblioteki	5
4.1.1	Cabal	5
4.1.2	Happstack	5
4.1.3	HUnit, QuickCheck, test-framework	5
4.1.4	postresql, sqlite3, HDBC	5

4.1.5	Parsec 2	5
4.1.6	xhtml 3000	5
4.1.7	inne (filemanip, strict, mtl)	5
4.2	Narzędzia	5
4.2.1	Git	5
4.2.2	Latex	5
4.3	Biblioteki	5
5	Słownik	5

1 Wprowadzenie

1.1 Cel dokumentacji

Celem niniejszego dokumentu jest takie przedstawienie struktury projektu, aby umożliwić jego modyfikacje oraz utrzymywanie programistom, którzy znają Haskell, ale nie należeli do początkowego zespołu. Wykaz użytych bibliotek powinien być pomocą, gdy instalacja oprogramowania nie powiedzie się i konieczna będzie kompilacja programu ze źródeł. Dodatkowo życzeniem autora jest nakreślenie wykonanej pracy tak, aby zainteresowane osoby były w stanie (w razie potrzeby) na wykorzystanie opisanych tu rozwiązań w swoich projektach.

2 Organizacja projektu

2.1 Ogólny opis

Program został napisany niemal w całości w Haskellu. Uruchamiany jest za pomocą wiersza poleceń, a komunikacja z użytkownikiem odbywa się za pomocą interfejsu WWW. Lista katalogów, które stanowią kolekcję plików SGF zapisywana jest do pliku konfiguracyjnego, wstępnie przetworzone (*znormalizowane*) gry są przechowywane w bazie danych. Dialog z użytkownikiem może odbywać się w języku polskim bądź angielskim.

2.2 Konfiguracja programu

Dane konfiguracyjne

2.2.1 Opis formatu pliku CONFIG

```
header:value1;value2;
```

2.3 Baza danych

Program pozwala na użycie baz Sqlite3 oraz PostgreSQL. Możliwa jest zmiana decyzji co do tego, która z nich jest używana; wymagane jest wówczas przebudowanie zawartości, gdyż protokół komunikacyjny bazy PostgreSQL nie jest kompatybilny z protokołem bazy Sqlite3. Ani użytkownik ani programista nie muszą zajmować się ręczną administracją bazy danych: służy do tego odpowiedni moduł (DB w pliku `src/DB.hs`).

Używana jest jedna tabela o nazwie `go_stat_data`.

2.3.1 Tabela go_stat_data

Opis pól tabeli go_stat_data

Pole	Typ	NULL dozwolone?	Opis
id	PRIMARY KEY	nie	unikatowy identyfikator gry
winner	CHAR	nie	zwycięzca gry ('b' lub 'w')
moves	VARCHAR(700)	nie	znormalizowany przebieg rozgrywki
path	VARCHAR(255)	nie	względna ścieżka do gry
b_name	VARCHAR(30)	nie	pseudonim (nazwisko) czarnego
w_name	VARCHAR(30)	nie	pseudonim (nazwisko) białego
b_rank	VARCHAR(10)	tak	ranking czarnego
w_rank	VARCHAR(10)	tak	ranking białego

Pola takie i takie.

2.4 Struktura modułów

Lista modułów wchodzących w skład projektu:

2.5 Pozostałe pliki

Pliki css, javascript, obrazki

3 Kompilacja i testowanie

4 Wykorzystane biblioteki i narzędzia

4.1 Biblioteki

4.1.1 Cabal

4.1.2 Happstack

4.1.3 HUnit, QuickCheck, test-framework

4.1.4 postgresql, sqlite3, HDBC

4.1.5 Parsec 2

4.1.6 xhtml 3000

4.1.7 inne (filemanip, strict, mtl)

4.2 Narzędzia

4.2.1 Git

4.2.2 Latex

4.2.3 Emacs

5 Słownik

Czarny

Biały

plik SGF

interfejs WWW

serwer HTTP

Haskell

zapis gry

ranking (gracza)

znormalizowany przebieg gry