Tuqqna

Sami Saada

Helsinki 5.10.2009

Design Document

HELSINGIN YLIOPISTO
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author

Sami Saada

Työn nimi — Arbetets titel — Title

Tuqqna

Oppiaine — Läroämne — Subject

Programming in Python

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Design Document	5.10.2009	3 pages

Tiivistelmä — Referat — Abstract

Design document of Tuqqna presents the design of Tuqqna, another implementation of Connect Four game. This document contains some information about the user interface, how it will be implemented. There is also section of components of project, which packages, modules and classes there will be.

Avainsanat — Nyckelord — Keywords

Python, Design document

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Exercise group teacher is Lauri Alanko (lealanko)

Sisältö

1 What is Tuqqna

Tuqqna is another implementation of Connect Four game, which is written in Python programming language. The name of Tuqqna comes from Kabyle word, which means connection. It will be fully tested, thanks to Test-Driven Development. The orginal reason to create Tuqqna is project for Programming in Python course.

1.1 What Kind of Game is Connect Four

Connect Four is well-known game, where two players drop buttons and try to connect four or more button in horizontal, vertical or diagonal lines. The basic Connect Four game board is 7x6, there are few variants of size. Tuqqna will be the basic 7x6, but that can be changed too.

Longer and specific description is in Connect Four article in Wikipedia: `http://en.wikipedia.org/wiki/Connect_Four`

1.2 What's Different

The basic game mechanic is same as in most of implementations of Connect Four. Board looks same and rules are same. There is no high scores, but Tuqqna counts victories and losses of both players in each game session. This provides opportunity to organize Connect Four tournament with Tuqqna. Tuqqna has own Artificial Intelligence to play against human players. Of course human versus human is always an option.

2 User Interface

There will be two user interfaces in Tuqqna, one to use in command line and one for graphical environment.

2.1 Command line User Interface

Command line user interface in Tuqqna will contain two parts. The first part is for information output, like help file. The second part contains curses user interface. Curses will be very handy interface for a game like Tuqqna, it provides all the functionality that is needed.

Curses is part of Python's standard library so it wont add more depencies. Many highly used command line application, like irssi, uses curses or ncurses, so it wont be bad choice. Curses contains lots of handy things like input areas and screen refreshing. Curses can also have mouse support, which isn't bad thing either.

2.2 Graphical User Interface

Graphical user interface in Tuqqna will be written with Tkinter. The reason of using Tkinter is that is de-facto standard GUI in python. Tuqqna will have graphical user interface, because it is the most common game interface in these days.

2.3 UI Information Description

Both of user interfaces contains lots of information. There can be selected players with names. Also an AI can be selected instead of another human player. UI will always show the current state of board. Player will always see all the things that it should see.

3 Components of Project

Tuqqna has lots of different kind of components.

3.1 Packages

Tuqqna will be build with nested packages. The main package is tuqqna and it has user interface packages, *core* and *ai* packages and unit tests.

User interface packages, *cli* and *gui*, contain all user interface specific modules.

Game core package, *core*, contains all core modules of Tuqqna.

Game artificial intelligence package, *ai*, contains all AI relevant modules.

All unit tests are inside of *test*-package. That package has subpackage for each package of Tuqqna and those packages contains unit test for its relevant game package.

3.2 Modules

There is huge amount of modules in Tuqqna. Each module contains relevant class(es) for that module. Every main feature of game will have own module: board, player, rules, etc.

3.3 Classes

Tuqqna is created with object-oriented programming, so classes have very important part to play.

All unit tests will be created in test case classes.