

Tactics

by We We

Submission date: 24-Mar-2022 04:04PM (UTC-0500)

Submission ID: 1792116869

File name: tictactoe_2.docx (74.62K)

Word count: 1625

Character count: 8949

Software Requirements Specification

for

Tic-tac-toe application

Student Name:

University:

Date:

2

Table of Contents

List of figures	2
1. Introduction	3
1.1 Proposed application	3
1.2 Objective of the proposed application	3
1.3 Scope of the application	3
1.4 System Specifications	4
1.5 Development tools and Programming Languages	4
1.6 Features	4
1.7 Expected Requirements.....	5
1.8 Risk Management	5
2.0 Technology Survey	6
2.1 JVM (Java Virtual Machine).....	6
2.2 Class libraries.....	7
2.3 JRE (Java Runtime Environment).....	7
3.0 Analysis and Modelling	8
3.1 Application strategy	8
3.2 UML Modelling	9
3.2.1 Sequence Diagram	9
3.2.2 Use case Diagram	10
3.2.3 Activity Diagram	10
3.3 Core Logic	11
3.4 Conclusion	11
References	12

4

List of figures

Figure 1: Sequence Diagram.....	9
Figure 2: Use case Diagram.....	10
Figure 3: Activity Diagram.....	10

1. Introduction

In development of a game project, the product is the game. For this project, it will be about development ¹ of a tic-tac-toe game application.

1.1 Proposed application

The Tic-tac-toe game application ¹ is a meant for two players, X and O, that take turns in marking a grid of either 3 by 3 or other grid sizes depending on the user. The game is a paper and pencil game. A player who is successful ¹ in placing the three particular marks in either a horizontal, vertical, or diagonal row is considered the winner of the game. The X player usually goes first. The game is commonly used to teach good sportsmanship concepts and to gain knowledge in artificial intelligence.

1.2 Objective of the proposed application

The main objective of the application is to develop a game application that will make a user attentive and alert in every situation they face. The application objective is to apply tic-tac-toe strategies that will never allow, or rather minimize, a loss, so the computer can either draw or win. If the user is not keen, then he/she will be hit by saucer-bombs for not noticing the saucer-fires. The application is also meant to be easy and simple, allowing even kids to play by using the neighboring keys on a keyboard.

1.3 Scope of the application

The application is meant to provide a virtual image of the combination of structured and non-structured ⁶ information of the tic-tac-toe application. The application is intended to run on the Windows platform. The player of the application will progress through the various levels that will require specific manipulations throughout the game environment that encourage creativity in making the moves along the pathways.

1.4 System Specifications

Most of the applications require high configurations of computers or mobile devices. Luckily, the configuration of the proposed application does not require a lot of system requirements. The minimum requirement of the proposed tic-tac-toe application is as follows:

- Operating system: Windows
- Processor: 1.2GHz
- RAM: 512 MB
- Storage: 200MB

1.5 Development tools and Programming Languages

- Microsoft Azure: provides a platform for application development using range of available technologies and programming languages.
- Java Development Kit: this will provide the appropriate tools for the development and testing of programs written in Java languages, and programs running on Java platform.
- Java: is an object-oriented programming language that produce software for multiple platforms.

1.6 Features

- Single player
- Multi-player
- Background music
- Different themes

1.7 Expected Requirements

The application requirements are important and so fundamental that the user of the application does not exactly require them to be stated. The absence of these requirements causes dissatisfaction, and they include:

- Maximum high definition
- Design the system within limited cost
- Minimum hardware requirements relevant to the application
- Design the whole application in an efficient manner

1.8 Risk Management

Risk is the uncertain event of an occurrence that can have either a positive or negative effect on the proposed application goals. It allows the programmer to achieve the required objectives of the proposed application, thus ensuring its successful completion. Possible risks to be encountered in the development of the proposed application include:

- The probability of the user's acceptance.
- Low quality probability of selecting the required requirements.
- Low probability of taking wrong decisions.
- Probability of not completing the application development within the limited time.
- Probability of selecting a technology component fit for the purpose of the application development.

2.0 Technology Survey

Since the proposed application is a web application developed using Azure, it will use Java as the programming language. (Wu, 2019), stated in his book that Java is a simple object-oriented language with strong exception-handling techniques for writing dynamic and extensible programs. Java language is used to write applications that are used to process words, playing games, storing data, or do a lot of things that a computer software is able to do.

Sun Microsystems developed Java language, and released in 1995. It derives most of its syntax from C and C++ languages, but with simpler and few low-level utilities. Java language creation has several goals attached to it. The goals include:

- Java language should use the object-oriented programming methodology.
- Programs developed using Java should be executed on multiple operating systems.
- Java language should contain in-built support for computer networks.
- Java language should be designed to remotely execute code.
- Java language should be easy to use.

The Java platform consists of several programs, each of which provides unique overall capabilities. The programs are the Java virtual machine, class libraries, and Java Runtime Environment.

2.1 JVM (Java Virtual Machine)

This is considered the Java platform heart, and is responsible for executing Java byte-code programs. The Java compiler translates the Java byte code and stores the code in memory during execution. The byte code is an intermediate language that allows all Java programs to run on any virtual machine available. Java Virtual Machine is available in every operating system.

2.2 Class libraries

Modern operating systems contain reusable code that simplifies the job of a programmer. The reusable code is a set of loadable libraries called at a runtime by most applications. Considering the applications cannot depend on the existing libraries, Java is considered not dependent on any specific operating system. The libraries provide the programmer with a set of well-known functions to perform common tasks. In addition, the class libraries provide a symbolic interface to the tasks that would heavily depend on the operating system and hardware. Finally, the libraries provide a compatible way to check for distinct features in an application.

2.3 JRE (Java Runtime Environment)

This is the software required for running applications deployed on Java platform. This allows the system not to crash as a result of exceptions (errors).

3.0 Analysis and Modelling

This chapter deals with the logical aspect of the application. Analysis deals with the comprehensive study of diverse operations performed by the application and their connection between within and outside the application. Available files are handled by the application. In this phase, the programmer studies the current application, which is a manual tic-tac-toe, and collects necessary data with regard to the application to be developed. According to a journal, (Software reliability modeling, 2019) the use of flowcharts and data flow diagrams comes in handy at the analysis and modeling phases as they indicate the flow of data in the application to be developed. The analysis of the proposed application is conducted using the objectives below in mind:

- Identifying of the user needs
- Assessing the application concepts for feasibility
- Performing economic and technical analysis

3.1 Application strategy

The number of states is finite; thus, we can enumerate the number of wins of both the computer and the user playing the tic-tac-toe application. When writing the application, we follow the following procedure:

- In the next step, avoid letting the application to enter a winning state for the user.
- Make a choice such that the application will transit to a winning state for the computer in the next step
- Finally, if such choices do not exist, pick a choice randomly

3.2 UML Modelling

This is a modeling language in software engineering intended to provide a way to visualize design of the application (Weilkiens, 2011).

3.2.1 Sequence Diagram

Considering the application is yet to be designed, the sequence diagram shows the exchange of information between the objects needed to carry out the application functionality. The diagram below shows an example of the sequence diagram of the proposed application.

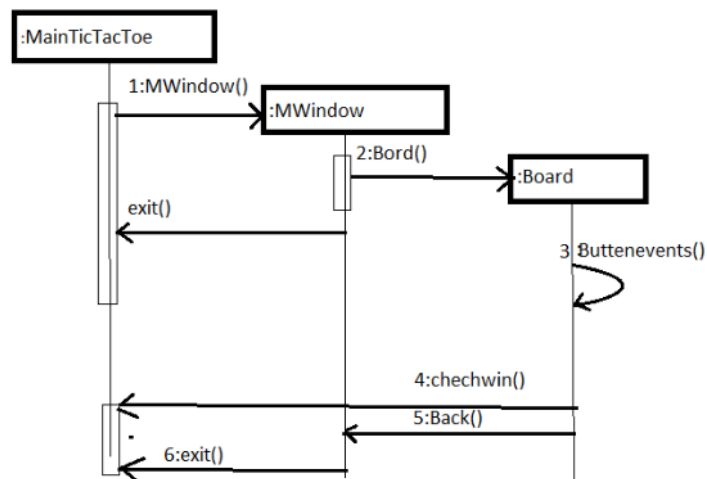
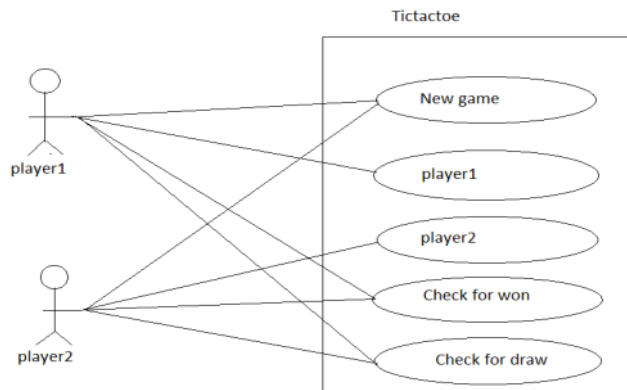


Figure 1: Sequence Diagram

3.2.2 Use case Diagram

This models the application's behavior and helps in capturing the application's requirements. The diagram below displays a use case example for the proposed application.



5
Figure 2: Use case Diagram

3.2.3 Activity Diagram

This shows the graphical representation of the application steps and activities with iteration. The diagram below shows an example of an activity diagram for the proposed application.

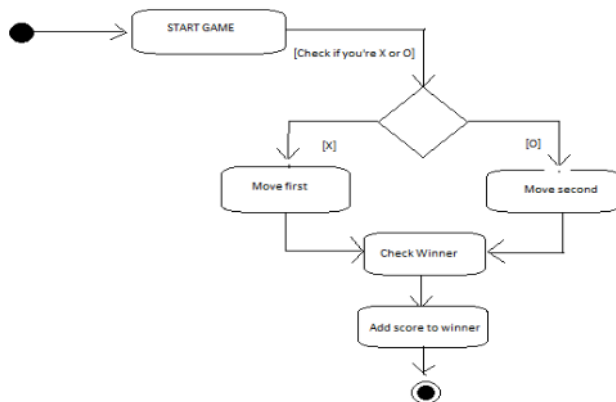


Figure 3: Activity Diagram

3.3 Core Logic

There are two logics in this tic-tac-toe application: when both players are humans and when one user is a computer. When the user playing is human, for each move, they check whether any three combinations are occupied by any player, and the winner is displayed.

3.4 Conclusion

The tic-tac-toe application is the most common game among all age groups. Application development has been suggested many times to curb the manual tic-tac-toe game that uses a pencil and paper. The proposed application will allow users to be creative and gain knowledge in the field of artificial intelligence. The proposed application is meant to be developed in the simplest way possible and have an interactive interface and other features to allow the application to be appealing and used by different users from all over the continent. The application to be developed should be free from bugs, and this is possible after it has been tested.

References

Software reliability modeling. (2019). *Modeling for Reliability Analysis*.

Weilkiens, T. (2011). *Systems engineering with SysML/UML, Modeling, analysis, design*.
Elsevier Publishers.

Wu, C. T. (2019). *An introduction to object-oriented programming with Java*. McGraw-Hill
Publishers.

Tactics

ORIGINALITY REPORT

5%

SIMILARITY INDEX

3%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1	Chanjuan Liu, Yuan Liu, Enqiang Zhu, Qiang Zhang, Xiaopeng Wei, Bin Wang. "Cross-Inhibitor: a time-sensitive molecular circuit based on DNA strand displacement", Nucleic Acids Research, 2020 Publication	2%
2	diva-portal.org Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	scholarworks.lib.csusb.edu Internet Source	1%
5	id.123dok.com Internet Source	1%
6	J. Muskens, R.J. Bril, M.R.V. Chaudron. "Generalizing Consistency Checking between Software Views", 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05), 2005 Publication	<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On