Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Term Project

CS319 Term Project: Disc-o-four

Analysis Report

Mehmet Emre Arıoğlu, Güneş Batmaz, Abdurrezak Efe, Enes Kavak

Supervisor: Bora Güngören

Progress Report

October 15, 2016

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object Oriented Software Engineering Project for course CS319.

Contents

[1 Introduction 1](#_Toc464340218)

[2 Overview 2](#_Toc464340219)

[2.1. Play Game 2](#_Toc464340220)

[2.2. Disc Types 2](#_Toc464340221)

[2.3. Bonuses 2](#_Toc464340222)

[3 Requirements 4](#_Toc464340223)

[3.1 Functional Requirements 4](#_Toc464340224)

[3.1.1 Play 4](#_Toc464340225)

[3.1.2 Game Options 4](#_Toc464340226)

[3.1.3 Settings 4](#_Toc464340227)

[3.1.4 Help 5](#_Toc464340228)

[3.1.5 About 6](#_Toc464340229)

[3.1.6 Exit 6](#_Toc464340230)

[3.2 Non-Functional Requirements 6](#_Toc464340231)

[3.2.1 Performance 6](#_Toc464340232)

[3.2.2 User Interface (UI) 7](#_Toc464340233)

[3.2.3 Open Source Code 7](#_Toc464340234)

[3.2.4 Extendibility 7](#_Toc464340235)

[4 System Model 8](#_Toc464340236)

[4.1. Use Case Model 8](#_Toc464340237)

[4.4.1. Play Game 8](#_Toc464340238)

[4.1.2 Settings 10](#_Toc464340239)

[4.1.3 Help 11](#_Toc464340240)

[4.1.4 About 12](#_Toc464340241)

[5 References 14](#_Toc464340242)

Analysis Report

CS319 Term Project: Disc-o-four

# Introduction

As a group, we would like to introduce you, Disc-o-four. To fulfill the needs of this lecture, we decided to implement a game as our term project, and from many different ideas, we thought that Disc-o-four will be a better choice since it can be joyful for everyone, in every age, who wants to play a board game.

We are inspired from Connect-Four game while we are creating this game. In this version of the classic game, there will be a game for 2 players as usual, and the players will compete with each other, and the first one who can connect four disks will win. In addition to the original game, we are planning to add time limitations, and bonuses to make the gaming experience more fun and interesting.

This game will be desktop application which is compatible for Microsoft Windows, and will be controlled by the keyboard.

In this Analysis Report, we will make detailed analysis of this game, and attempt to create a well analysed game.

# Overview

## Play Game

The game starts as the computer decides which player will starts randomly. Then, the selected player puts his/hers first disc which is also assigned according to the disc colours which are specified in the Section 2.2. .Then the game goes on until one of the players puts a disk on the bonus areas or connects four disks in their colour. When the player puts a disk on the bonus area, one of the bonuses, which are explained in Section 2.3. in detail, assigns to the player, and the game goes on until one of the players connects four disc. The game is played by the arrow-keys and A-S-D-W keys by the player.

## Disc Types

In Disc-o-four, there will be four types of discs and their meanings are as follows:

***Yellow Discs:*** This disc represents the discs of the Player 1

***Orange Discs:*** This disc represents the discs of the Player 2

***Lilac Discs:*** This disc represents the discs of the Joker Disc, in other words, the disk which can be used by the both players.

***Black Discs:*** This disc represents the discs of the Blocking Disc, in other words, the disk which cannot be used by the any of the players.

## Bonuses

In Disc-o-four, there will be bonus areas which are determined before the game randomly by the computer, and shown as the game starts. When a player puts a disc on the specific are, the computer, again randomly, assigns a bonus, which can be a good or a bad bonus.

**Delete Bonus:** The player who put his/hers disc into the bonus area can delete an unwanted disc.

**Extra-Move Bonus:** The player who put his/hers disc into the bonus area can make an additional move.

**Loosing-Turn Bonus:** The player who put his/hers disc into the bonus area will lose his/hers turn, so the other player takes the next turn.

**Computer’s Turn Bonus:** Instead of the player, computer makes a random move, and in this move, it will put either one of the Lilac or Black (the characteristics of them are given in Section 2.2.) disc.

# Requirements

## 3.1 Functional Requirements

### 3.1.1 Play

This function is the first Menu button operation that leads to the opening screen of the game itself. This function opens a Game Object sample and pops up a game starting option screen.

### 3.1.2 Game Options

This function is called by the Play function and helps user to determine the attributes that the Settings function (will be examined in following lines) will not change in any other place. In other words, Settings function will change the attributes that are to be permanent in following games; meanwhile, this function changes the attributes of the current game.

Game Options function will help users to determine the upcoming attributes:

* User names (Players decide on their names)
* The player to start (The computer chooses randomly)

### 3.1.3 Settings

This function opens a screen that leads through user to change some of flexible features such as;

* Change Background Music
* Change Background Picture
* Music(On/Off)
* Effect Sound(On/Off)

#### 3.1.3.1 Change Background Music

The game has a music library that can only be manipulated by the user from its file directory. Nevertheless, in the game the user can pick a particular music from the existing ones that are in the library. The chosen one will be played on the background while playing.

#### 3.1.3.2 Change Background Picture

There are some pictures of high definition for users to choose for the background of the game screen. As the game screen will be transparent in terms of lines and coordinates, the background picture will be a classy attribute. Like the Background Music attribute, the Background Picture is also changeable using its file directory.

#### 3.1.3.3 Music (On/Off)

This function, as its name proposes, is to set On or Off the music feature.

#### 3.1.3.4 Effect Sound (On/Off)

This function is to stop or start the effect sounds such as Disc falling, Bonus win, Anti-Bonus, Game win etc.

### 3.1.4 Help

Help function opens a screen that has links to a quick tutorial of the game, rules of it and extra features. The main titles are:

* How to Play (Mostly UI)
* Rules and Extras

#### 3.1.4.1 How to Play

As the name points out, this function leads into a screen that mentions which keys to move, let the disc fall etc.

#### 3.1.4.2 Rules and Extras

This function guides user to a screen that explains the rules and gives detailed information about the bonuses, anti-bonuses and how they work.

### 3.1.5 About

The About function escorts users to a page that includes the whole information about the developers, the game’s history, mathematical background of the game (the winning strategy) and a link to the project’s GitHub repository. The page will also have the developers’ contact information.

### 3.1.6 Exit

This function shuts down the process.

## 3.2 Non-Functional Requirements

### 3.2.1 Performance

The performance is a big case as of 21st century in terms of both memory and time. We intend to design the program so that it will have a disk threshold (the music and the pictures will be limited) and low RAM cost per second.

### 3.2.2 User Interface (UI)

The user interface is going to be designed with only one principle. It will be pure and plain! The users will not be distracted by anything with our friendly UI.

### 3.2.3 Open Source Code

The whole project code will be published in the GitHub repository so that developers who are interested in enhancing our design or people who wants to create something similar would be able to use it!

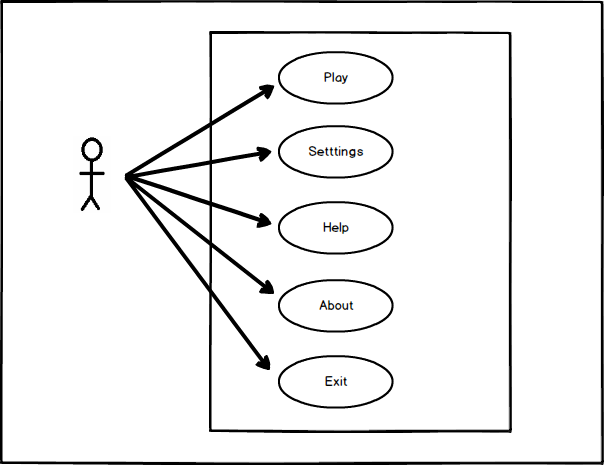
### 3.2.4 Extendibility

This project is like a ladder to the upcoming projects which could include the game or an updated version. For instance, if the time allows us we will be working on an AI for the game to provide users with a single player game against the computer. Additionally, as the source code will be public, it will be always developing itself.

# System Model

## Use Case Model

The following section explains the use case model of Connect Four. Detailed information about the game are listed below.

****

**Figure: Illustration of Connect Four case model**

### 4.4.1. Play Game

**Actors:** Player 1 and Player 2

**Stakeholders and Interests:**

Each player tries to make a series of four disk to win  
System keep time.  
  
**Pre-conditions:** Initially game settings are set as default. If player change the game settings it will be applied.  
**Post-condition:** The end-game screen will show up if either of the players win.  
  
**Entry Condition:** Player clicks “Play Game” on Main Menu.  
**Exit Condition:** Player clicks “Back” to return to Main Menu.  
  
**Success Scenario Event Flow:**

1. Game is started with default settings.

2. Player 1 starts.

3. Player 2 starts.

4. Players play the game as turn-based and when game is finished when one wins.  
5. Final screen shows up and if “Play Again” is clicked, game starts again else it returns to Main Menu.

**Alternative Flows:**  
A. If any of the players wants to give up:  
 A.1. Player selects “Give Up” in his/her turn.  
 A.2. System returns to Main Menu.

B. If there is no winner (i.e. tie)

B.1. Final screen shows up.

B.2. New game starts if “Play Again” is selected.

B.3. Returns to Main Menu if “Back” is selected.

C. If any of the players do not make a move in the given time limit.

C.1. Current player loses.

C.2. Final screen shows up.

C.3. New game starts if “Play Again” is selected.

C.4. Returns to Main Menu if “Back” is selected.

### 4.1.2 Settings

**Actors:** Player 1 and Player 2

**Stakeholders and Interests:**

Players wish to change default game settings:

-Changing background.

-Enabling or disabling the game sounds.

Settings that are changed by the players are updated by the system.

**Pre-conditions:** Game settings are set as default and adjusted if players make any changes.  
**Post-condition:** Changed game settings are updated.  
  
**Entry Condition:** Players are on Main Menu and choose “Settings”.  
**Exit Condition:** Players return to Main Menu.  
  
**Success Scenario Event Flow:**

1. Players click “Settings” button to make changes.
2. System displays the game settings screen.
3. Players change the defaults according to their desire.
4. Changes are updated by the system successfully.

**Alternative Flows:**  
A. If players wish to return to Main Menu:

A.1. Players choose “Back” to return.

A.2. System shows Main Menu.

### 4.1.3 Help

**Actors:** Player 1 and Player 2

**Stakeholders and Interests:**

- Players want to learn how to play Connect Four.  
-The rules of the game and an example of how to play the game is shown by the system.

**Pre-conditions:** Players must be in the Main Menu.  
**Post-condition: -**  
  
**Entry Condition:** Players choose “Help” form Main Menu.  
**Exit Condition:** Players return to Main Menu.  
  
**Success Scenario Event Flow:**

1. System shows the game rules and how-to-play Connect Four.

**Alternative Flows:**  
A. If players wish to return to Main Menu:

A.1. Players choose “Back” to return.

A.2. System shows Main Menu.

### 4.1.4 About

**Actors:** Player 1 and Player 2

**Stakeholders and Interests:**

Players want to see the team who developed Connect Four and information about them.

Names and contact information of the developers are shown by the system.

**Pre-conditions:** Players must be in Main Menu.  
**Post-condition: -**  
  
**Entry Condition:** Players choose “About” on the Main Menu.  
**Exit Condition:** Players return to Main Menu by choosing “Back”.  
  
**Success Scenario Event Flow:**

1. Names and information about the developers of Connect Four is listed.

**Alternative Flows:**  
A. If players wish to return to Main Menu:

A.1. Players choose “Back” to return.

A.2. System goes back to Main Menu.

Scenario #1: Executing Connect Four

X opens Connect Four by clicking on the game icon from the disk. Main Menu appears and X selects “Settings”. After changing background and disabling sounds, she clicks on “Back” to return to Main Menu. The she clicks on “Play Game” button to start the game.

Description:

After X executes the game, settings, help and about files are loaded from the disk. Next, the system is waiting for the following instruction. It can be any of the five options (“Play Game”, “Settings”, “Help”, “About”, “Exit”) on the Main Menu. In the current scenario, players choose “Settings” and “Play Game”. When “Settings” is clicked the background images that are stored in the disk will be fetched.

Scenario #2: Play Game

X and Y play the game. When X tries to put a disk into a space, the game does not let him do that. Instead, game itself puts a grey disk into that place which is called neutral. It has no good for any players. It is a difficulty of the game.

Description:

After the first player presses action button to fill the board, main loop starts. At the end of each turn, loop updates the board and checks the conflicts and whether there is any series of four disks with the same color or not. The objects are drawn to the board according to the game loop. A final window comes up when the winner is set.

Scenario #3: Getting bonus

Players has already started the game. Player 2 tries to put a disk into a particular space. Game blocks him and puts a green disk in there. That green disk helps both of the players. With the help of it, Player 2 makes a series of four and wins the game.

Description: When a player tries to put a disk in a space, bonus function is called and bonus is created. When a player takes the advantage of the bonus or not, necessary updates are made.

# References

1. Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.