Programozási technológia   
2. Beadandó

Név: Magyar Viktor

Neptun kód: O9WEJJ

Dátum: 2024.04.01.

Feladat száma: 1

[1. Feladat leírása: 4](#_Toc162851811)

[2. Megoldási terv: 6](#_Toc162851812)

[2.1 Megoldás leírása 6](#_Toc162851813)

[2.2 Model komponens 7](#_Toc162851814)

[2.2.1 tictactoe.table package: 7](#_Toc162851815)

[A képen szöveg, nyugta, diagram, Betűtípus látható

Automatikusan generált leírás2.2.2 tictactoe.gamecontrol package: 7](#_Toc162851816)

[A képen szöveg, nyugta, diagram, Párhuzamos látható

Automatikusan generált leírás2.2.3 tictactoe.gamemenu package: 7](#_Toc162851817)

[A képen szöveg, diagram, nyugta, Párhuzamos látható

Automatikusan generált leírás2.2.4 tictactoe.ObjectCompositionUtils 8](#_Toc162851818)

[A képen szöveg, Betűtípus, képernyőkép, szám látható

Automatikusan generált leírás 8](#_Toc162851819)

[2.4 Controller komponens 9](#_Toc162851820)

[2.4.1 tictactoe.applicationstates package 9](#_Toc162851821)

[A képen szöveg, nyugta, képernyőkép, Párhuzamos látható

Automatikusan generált leírás 9](#_Toc162851822)

[2.4.2 tictactoe.exceptions: 9](#_Toc162851823)

[A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás 9](#_Toc162851824)

[2.5 View komponens 10](#_Toc162851825)

[2.5.1 tictactoe.views package osztályai: 10](#_Toc162851826)

[2.5.2 tictactoe.views.mainmenu package: 11](#_Toc162851827)

[2.5.3 tictactoe.views.initializegameview: 11](#_Toc162851828)

[2.5.4 tictactoe.views.tictactoeview: 12](#_Toc162851829)

[Tesztelési terv: 13](#_Toc162851830)

# 1. Feladat leírása:

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy n × m mezőből álló tábla (n az oszlopok, m a sorok száma), amelyre a játékosok X, illetve O jeleket potyogtatnak (azaz egy adott oszlopban a karakter mindig „leesik” a legalsó üres sorba, függetlenül attól, melyik sorban helyeztük le). A játékosok felváltva lépnek, és egy oszlopban csak akkor helyezhetnek el új jelet, ha az még nem telt meg. A játékot az nyeri, aki előbb elhelyez vízszintesen, vagy átlósan négy szomszédos jelet. A játék döntetlennel ér véget, ha betelik a tábla. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (8×5, 10×6, 12×7), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hogy melyik játékos győzött (ha nem lett döntetlen), majd kezdjen automatikusan új játékot.

# 2. Megoldási terv:

## 2.1 Megoldás leírása

A feladat megoldását az MVC architektúra alapján készítettem el. Ami azt jelenti, hogy a program szétválasztható három komponensre, ami a Model, a View és a Controller. Részletesen bemutatom a következő fejezetekben, hogy mely package-ek, illetve osztályok teszik ki a három komponenst.

## 2.2 Model komponens

### 2.2.1 tictactoe.table package:

### A képen szöveg, nyugta, diagram, Betűtípus látható Automatikusan generált leírás2.2.2 tictactoe.gamecontrol package:

### A képen szöveg, nyugta, diagram, Párhuzamos látható Automatikusan generált leírás2.2.3 tictactoe.gamemenu package:

### A képen szöveg, diagram, nyugta, Párhuzamos látható Automatikusan generált leírás2.2.4 tictactoe.ObjectCompositionUtils

## A képen szöveg, Betűtípus, képernyőkép, szám látható Automatikusan generált leírás

## 2.4 Controller komponens

### 2.4.1 tictactoe.applicationstates package

### A képen szöveg, nyugta, képernyőkép, Párhuzamos látható Automatikusan generált leírás

### 2.4.2 tictactoe.exceptions:

### A képen szöveg, képernyőkép, Betűtípus, sor látható Automatikusan generált leírás

## 2.5 View komponens

A képen szöveg, diagram, nyugta, Párhuzamos látható

Automatikusan generált leírás

### 2.5.1 tictactoe.views package osztályai:

.

### 2.5.2 tictactoe.views.mainmenu package:

A képen szöveg, képernyőkép, Betűtípus, nyugta látható

Automatikusan generált leírás

.

### 2.5.3 tictactoe.views.initializegameview:

A képen szöveg, képernyőkép, Párhuzamos, szám látható

Automatikusan generált leírás.

### 2.5.4 tictactoe.views.tictactoeview:

A képen szöveg, képernyőkép, nyugta, Betűtípus látható

Automatikusan generált leírás.

# Tesztelési terv:

|  |  |  |  |
| --- | --- | --- | --- |
| **Név** | **Leírás** | **Bemeneti fájl** | **Kimeneti fájl** |
| Greedy\_Test | Teszteli a „mohó” játékos teljes logikáját. | Input1.txt | Output1.txt |
| Tactician\_Test | Teszteli a „taktikus” játékos teljes logikáját. | Input2.txt | Output2.txt |
| Cautious\_Test | Teszteli a „óvatos” játékos teljes logikáját. | Input3.txt | Output3.txt |
| Service\_Field\_Test | Teszteli a „szolgáltatás” típusú mező logikáját. | Input4.txt | Output4.txt |
| Luck\_Field\_Test | Teszteli a „szerencse” típusú mező logikáját. | Input5.txt | Output5.txt |
| Property\_Owned\_  By\_Player\_without  \_house\_and\_other\_  player\_steps\_on\_it | Teszteli, hogy egy játékos által megvett ingatlanra (ház nélküli), ha rálép egy másik játékos, akkor tényleg 500 petákot von-e le a másik játékostól. | Input6.txt | Output6.txt |
| No\_Input\_File\_Found | Teszteli, hogy mit ír ki a program, amennyiben nem található input fájl. | - | Output7.txt |
| Not\_Enough\_Random\_  Test\_Number\_In\_Input  \_File | Teszteli, hogy mit ír ki a program, ha nincs elegendő „kockadobás” a megadott input fájlban, tehát, ha nincsen vége a játéknak, mikor már elfogytak az input fájlban lévő „kockadobások”. | Input8.txt | Output8.txt |
| Invalid\_Input\_File\_  Negative\_Luck | Teszteli, hogy mit ír ki a program, amennyiben negatív számot adunk meg az egyik Luck típusú mezőnek. | Input9.txt | Output9.txt |
| Invalid\_Input\_File\_  Negative\_Service | Teszteli, hogy mit ír ki a program, amennyiben negatív számot adunk meg az egyik Service típusú mezőnek. | Input10.txt | Output10.txt |
| Property\_Owned\_  By\_Player\_without  \_house\_and\_owner\_  player\_steps\_on\_it | Teszteli, hogy egy játékos által megvett ingatlanra (ház nélküli), ha rálép a tulajdonos, akkor nem von-e le 500 petákot a játék. | Input11.txt | Output11.txt |
| Property\_Owned\_By\_  Player\_with\_house\_and\_  other\_player\_steps\_on\_it | Teszteli, hogy egy játékos által megvett ingatlanra (házzal), ha rálép egy másik játékos, akkor tényleg 2000 petákot von-e le a másik játékostól. | Input12.txt | Output12.txt |
| Property\_Owned\_By\_  Player\_with\_house\_and\_  owner\_player\_steps\_on\_it | Teszteli, hogy egy játékos által megvett ingatlanra (házzal), ha rálép a tulajdonos, akkor nem von-e le 2000 petákot a játék. | Input13.txt | Output13.txt |
| Invalid\_Input\_File\_Not\_  All\_Field\_Types | Teszteli, hogy mit ír ki a program, amennyiben nincsen minden típusú mezőből legalább egy darab. | Input14.txt | Output14.txt |