# Kamisado Game Komponens Dokumentáció

## Tóth Tamás

## 2014

Tartalomjegyzék

1. Kamisado 2

2. Komponens specifikáció 3

2.1. Struktúrális modell 3

2.2. Funkcionális modell 4

2.3. Viselkedési modell 4

3. Komponens realizálás 5

3.1. Struktúrális modell realizálása 5

3.2. Funkcionális modell realizálása 5

4. Implementációs technológia 6

5. Teszt esetek 6

5.1. Funcionális teszt 6

5.2. Unit teszt 6

# 1. Kamisado

Kétszemélyes, absztrakt stratégiai játék. A játék alapja a megfelelően színezett 8 sorból és 8 oszlopból álló tábla. A játékosoknak nem csak térben, hanem színekben is kell gondolkodniuk.

A rendelkezésre álló bábukkal előre és átlósan léphetnek felváltva a játékosok, a lépési szabályok betartásával.

A következő játékosnak mindig olyan színű bábujával kell lépni amilyen mezőre ellenfele utoljára lépett.

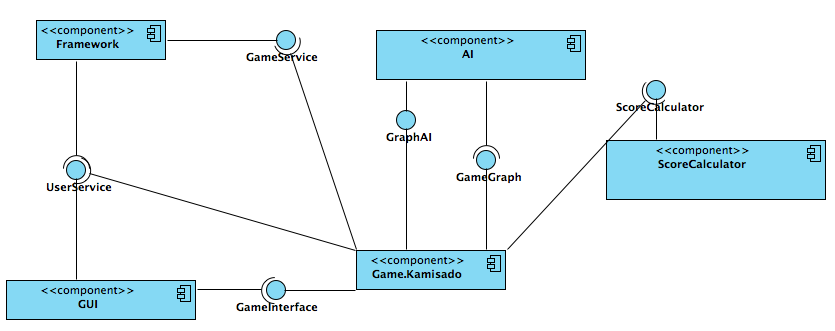
A játék akkor ér véget, ha az egyik játékos eljut a másik játékos kezdőpontjára.



# 2. Komponens specifikáció

A Game.Kamisado komponens valósítja meg a kamisado nevű játék logikáját és minden játékspecifikus funkciót. A komponens az előre meghatározott interfaceken keresztül kommunikál a többi komponenssel.

## 2.1. Struktúrális modell



A Game.Kamisado a meghatározozz interfaceken keresztül a következő kommunikál.

**Framework**

A perzisztens objektumok kezeléséért(felhasználók, játékállások) és a komponensek összekapcsolásáért felel.

**GUI**

Az általános megjelenésért felelős komponens, mely tartalmazza a felhasználók kezelésének felületeit is.

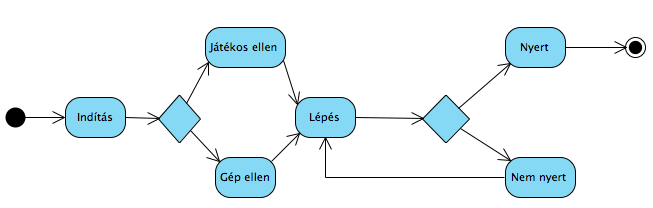
**AI**

A mesterséges intelligenciát megvalósító komponens, mely a játékos lépését kiértékelve adja meg a szükséges válaszlépést.

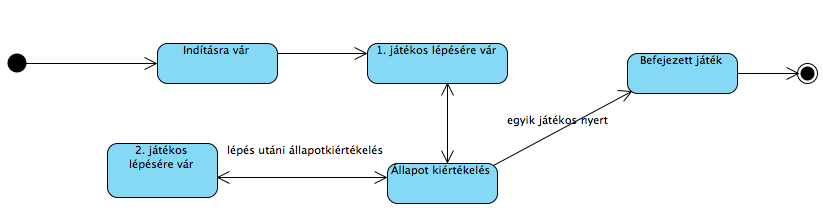
**ScoreCalculator**

Pontszámítást végző heurisztika, az aktuális állást kiértékelve egy pontszámot határoz meg melynek segítségével a különböző AI-k értékelhetik a játékot.

## 2.2. Funkcionális modell



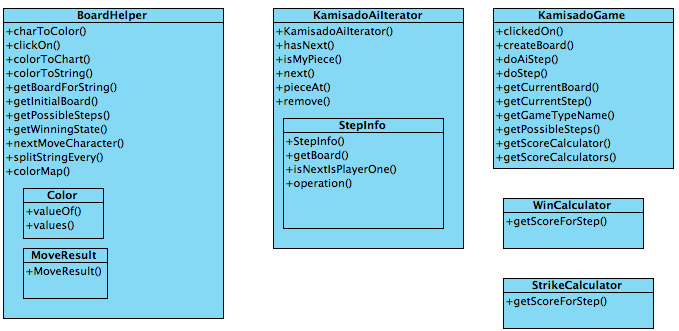
## 2.3. Viselkedési modell



# 3. Komponens realizálás

## 3.1. Struktúrális modell realizálása

A korábban meghatározott funkcionális és struktúrális specifikáció szerint készült a komponens.



## 3.2. Funkcionális modell realizálása

A KamisadoGame osztály végzi a komponens vezérlését.

A createBoard metódus segítségével a komponens létrehozza a játéktáblát. Az aktuális tábla állását a getCurrentBoard segítéségével kérhetik le a kapcsolódó komponensek.

Megkülönböztetjük a felhasználó és a gép által történő lépéseket, melyeket doAiStep és doStep metódusokban kezelünk.

A BoardHelper osztály tartalmazza a játéktábla feldolgozásához szükséges segédmetódusokat és alapértelmezett kiindulási értékeket, színeket.

# 4. Implementációs technológia

A komponens megvalósítása J2EE platformon történt, Enterprise JavaBeans felhasználásával, a NetBeans IDE segítségével.

A komponens közvetlenül nem csatlakozik a MySQL alapú adatbázishoz, ennek kezelését egy másik komponens végzi.

A komponens elsősorban a játék logikájáért felel, a megjelenítés szintén egy másik komponens feladata amely XHTML és CSS segítségével rajzolja ki a megfelelő játékot.

A megjelenítéshez szükséges definíciókat, színeket a komponensben tároljuk.

# 5. Teszt esetek

## 5.1. Funcionális teszt

A komponens tesztelését elsősorban funkcionális tesztek segítségével végeztem el, melyet az elkészült játék komponens elindításával és az egyes felhasználási esetek végigpróbálásával végeztem el.

## 5.2. Unit teszt

A funkcionális tesztek mellett létrehoztam jUnit teszt osztályokat az elkészült osztályok tesztelésére:

* KamisadoAiIteratatorTest
* KamisadoGameTest
* BoardHelperTest