

Toolkit In OPen source for Critical Applications and SystEms Development

Model AmobaRobotModel

This document has been generated using **gendoc2** : More information available here :

<http://www.topcased.org/index.php>

<http://marketplace.eclipse.org/content/gendoc2>

|  |  |  |  |
| --- | --- | --- | --- |
|  | Author(s) | Checked by | Approbation |
| Name |  |  |  |
| Compagny |  |  |  |
| Department |  |  |  |
| Date |  |  |  |
| Visa |  |  |  |
| summary |  |  |  |

|  |
| --- |
| **Attention** : la responsabilité des entreprises et des organismes ayant participé à l'élaboration de ce document ne peut en aucun cas être engagée en cas de dommages ou de pertes résultant de l'utilisation ou de l'exploitation des informations qui y sont contenues.  **Disclaimer** : Contractors participating to this report shall incur no liability whatsoever for any damage or loss which may result from the use or exploitation of information and/or Rights contained in this report. |

Sommaire

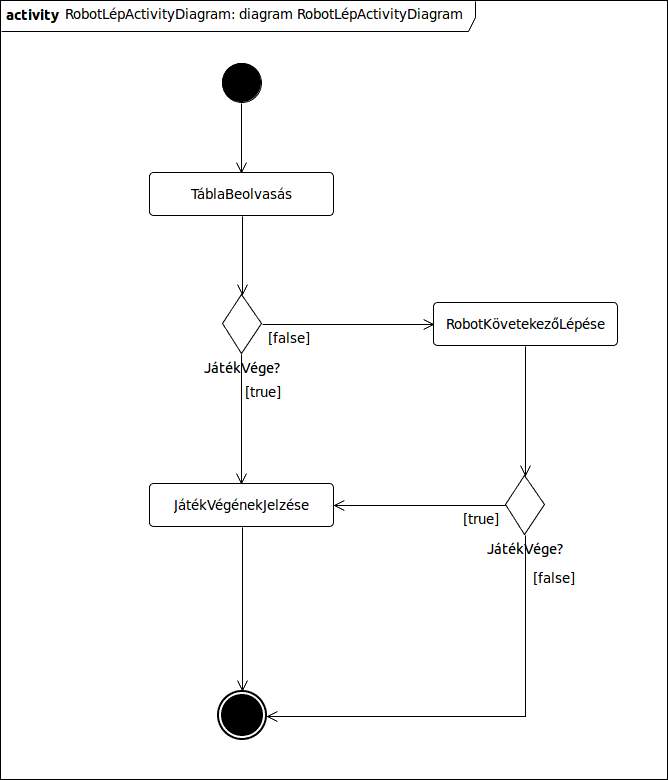
[1 Diagram : [d.name/] 3](#_Toc289330255)

[1.1 [clean(getText(e))/] 3](#_Toc289330256)

# Diagram : RobotLépActivityDiagram

From Activity RobotLépActivityDiagram

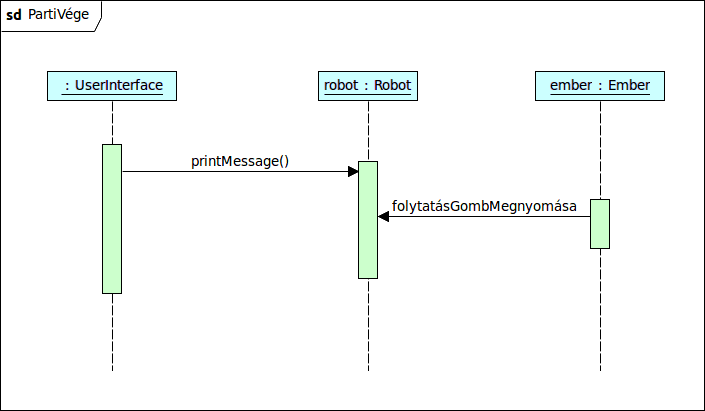
)/]' maxW='true' keepH='false'>



# Diagram : PartiVége

From Interaction PartiVégeDetail

)/]' maxW='true' keepH='false'>



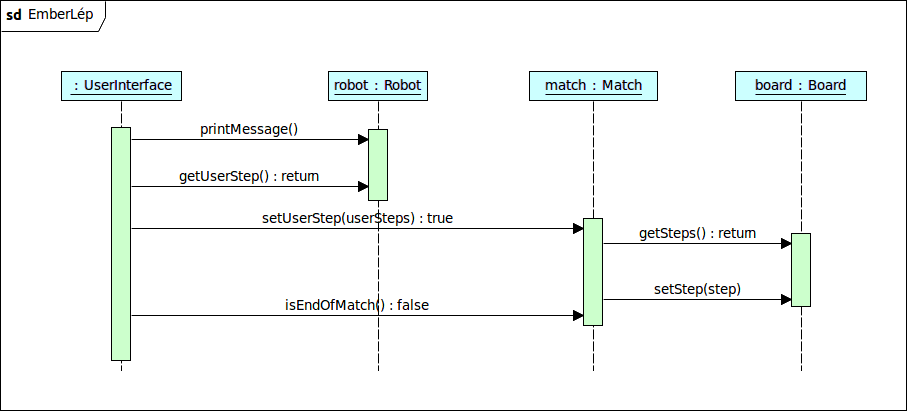
## Message printMessage

Az alábbiakat írja ki:
kiírja hogy ki nyert
"Szedd le a golyókat és nyomd meg a folytatás gombot."

# Diagram : EmberLép

From Interaction EmberLépDetail

)/]' maxW='true' keepH='false'>



## Message printMessage

Kiírja: "Lépjen"

# Diagram : ComponentView

From Component Main

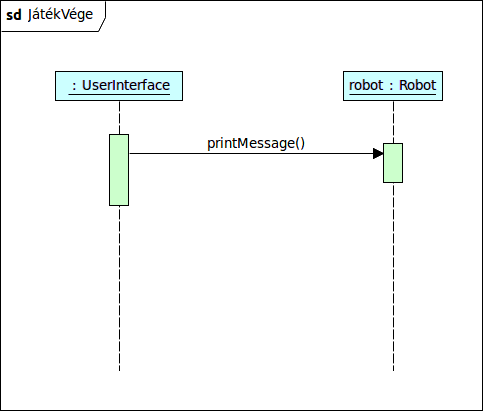
)/]' maxW='true' keepH='false'>



# Diagram : JátékVége

From Interaction JátékVégeDetail

)/]' maxW='true' keepH='false'>



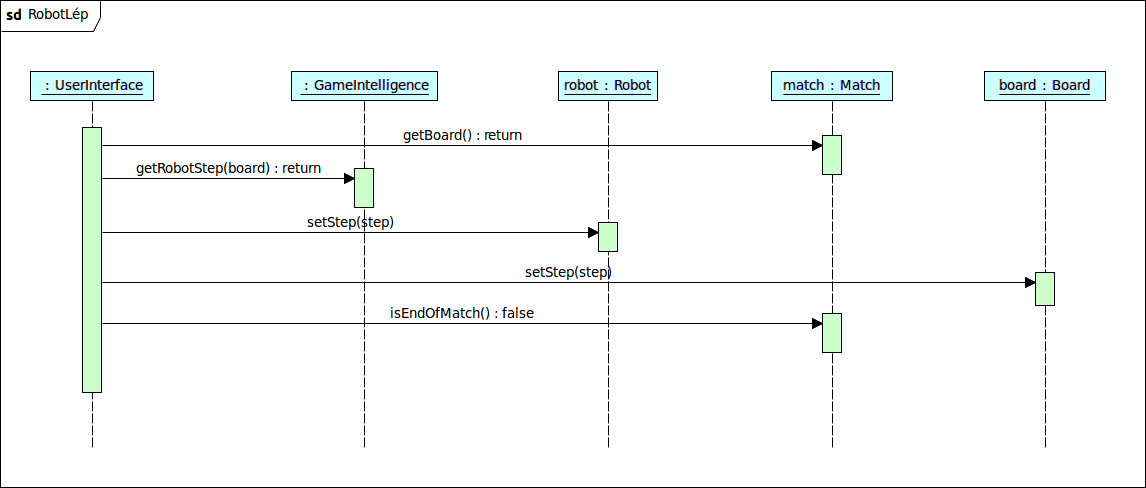
## Message printMessage

Kiírja a játék végeredményét.

# Diagram : RobotLép

From Interaction RobotLépDetail

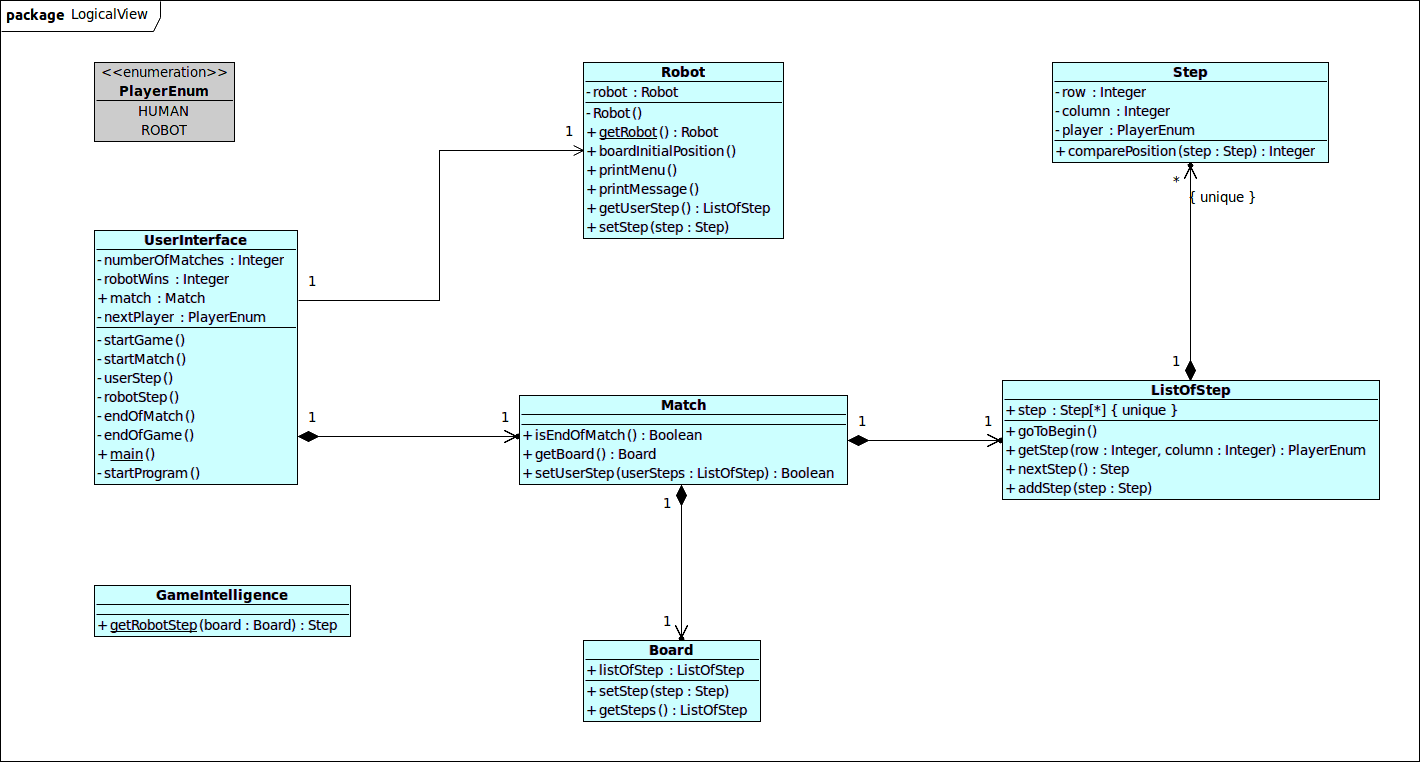
)/]' maxW='true' keepH='false'>



# Diagram : LogicalView

From Package LogicalView

)/]' maxW='true' keepH='false'>



## Class Robot

Ez az osztály felelős a robot fizikai irányításáért.
Ez Singleton kell legyen.

## Class Match

Ez reprezentálja a parti logikai egységet.

## Operation goToBegin

A lista eljére ugrik.

## Operation startMatch

UC06
match = new Match()
robot.printMessage("Parti indul")

## Property player

Játékos

## Operation comparePosition

Ez a függvény egy rendezési relációt definiál a Step objektumok között. A rendezéshez a row és column értékeket használja.
Ha a this < step, akkor -1,
ha a this = step, akkor 0,
ha a this > step, akkor 1 a visszatérési érték.

## Class ListOfStep

A lépések egy rendezett listáját valósítja me. A listát a getStep és nextStep eljárással csak egy konkrét rendezés szerint lehet bejárni.

## Operation setStep

Egy játékos lépés beállítása.

## Operation getRobotStep

Meghatározza a robot lépését.

## Operation startProgram

Ez írja ki a főmenüt, és kéri be a választ.

## Class Board

A játéktér absztrakt reprezentációja.

## Operation startGame

UC02
robot.printMenu(...) - Partik száma?
numberOfMatches értékének beállítása.
robot.printMenu(...) - Ki kezdje a játékot?
nextPlayer értékének beállítása.

## Operation endOfGame

UC05
// kiírja az eredményt

## Class UserInterface

Ez az osztály valósítja meg a felhasználói eseteket.

## Property row

Sor

## Operation main

startProgram()
while (nemExit) {
startGame()
for (int i=0;i<numberOfMatches;i++) {
startMatch()
while ( ! match.isEndOfMatch() ) {
if (nextPlayer = HUMAN) {
userStep()
} else {
robotStep()
}
}
endOfMatch()
}
endOfGame()
startProgram()
}

## Operation getBoard

Ezzel a metódussal lehet elkérni a játékteret reprezentáló objektumot.

## Operation getSteps

Az összes lépést egy rendezett listában adja vissza.

## Operation endOfMatch

UC07
// kiírja hogy ki nyert
// szedd le a golyókat és nyomd meg a kezdés gombot.

## Operation userStep

UC03
robot.printMessage("Lépjen!")
// beolvassuk a táblát
readedSteps = robot.getUserStep()
if (match.setUserStep(readedSteps))
nextPlayer = ROBOT
}

## Operation setUserStep

Az emberi játékos lépésének a meghatározását végzi el. Érvényes lépés esetén azt le is menti a Board-ba.
====================
// elkérjük az eddigi lépéseket
oldSteps = board.getSteps()
// elemenkénti feldolgozással egyszerre
// feldolgozzuk a userSteps és oldSteps listákat.
// Csalás esetén exception-t dobunk

## Operation isEndOfMatch

Ezzel a metódussal lehet ellenőrizni, hogy vége van e a partinak.

## Operation getStep

Visszaadja azt a játékost, akinek a bábuja az adott pozícón van. Ha nincs ott bábu, akkor null a visszatérési érték.

## Class GameIntelligence

Ez az osztály felel a robot lépésének a meghatározásáért.
Ez egy static osztály egy publikus metódussal.

## Operation robotStep

UC04
step = GameIntelligence.getRobotStep(board)
robot.setStep(step)
board.setStep(step)
nextPlayer = HUMAN

## Class Step

Egy lépés absztrakt reprezentációja.

## Operation nextStep

A lista bejárására szolgáló eljárás, amely visszaad egy Step objektumot, vagy null-t, ha a lista végetért.

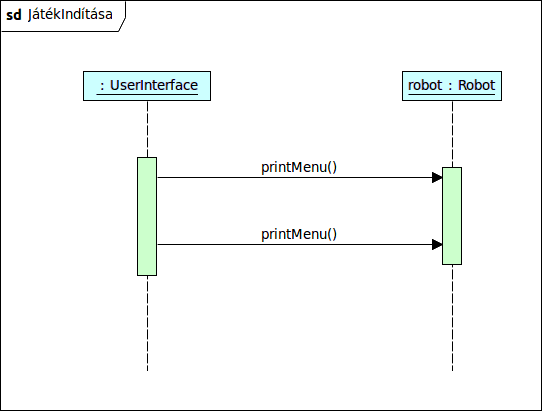
## Property column

Oszlop

# Diagram : JátékIndítása

From Interaction JátékIndításaDetail

)/]' maxW='true' keepH='false'>



## Message printMenu

Kezdő játékos bekérése.

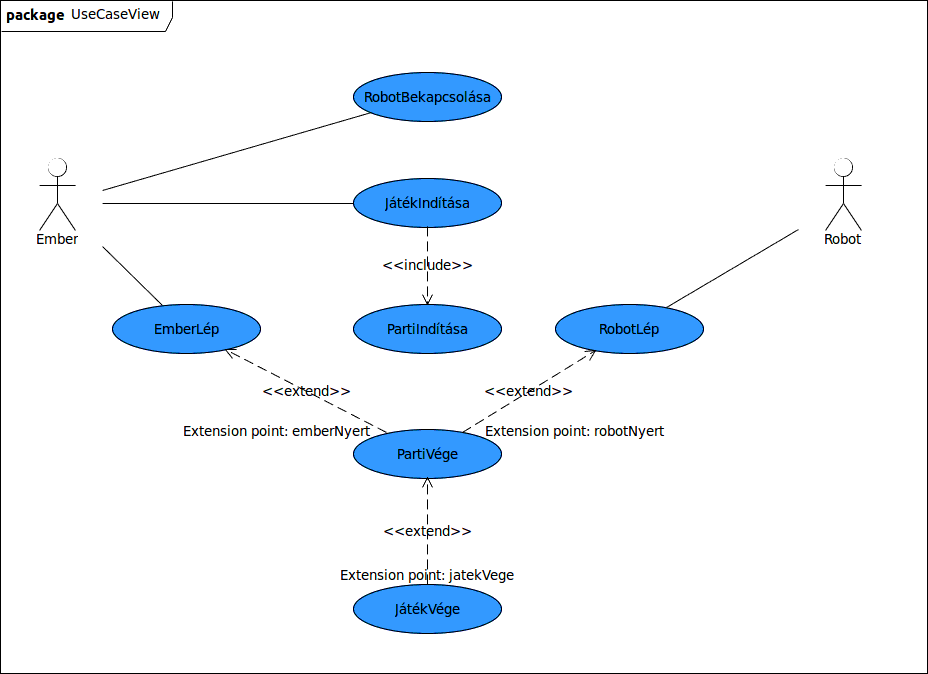
## Message printMenu

Partik számának bekérése.

# Diagram : UseCaseView

From Package UseCaseView

)/]' maxW='true' keepH='false'>



## Use Case JátékVége

UC05

## Use Case RobotLép

UC04

## Use Case PartiIndítása

UC06

## Use Case JátékIndítása

UC02
Parti szám kiválasztása
Játék indítása

## Use Case EmberLép

UC03

## Use Case RobotBekapcsolása

UC01

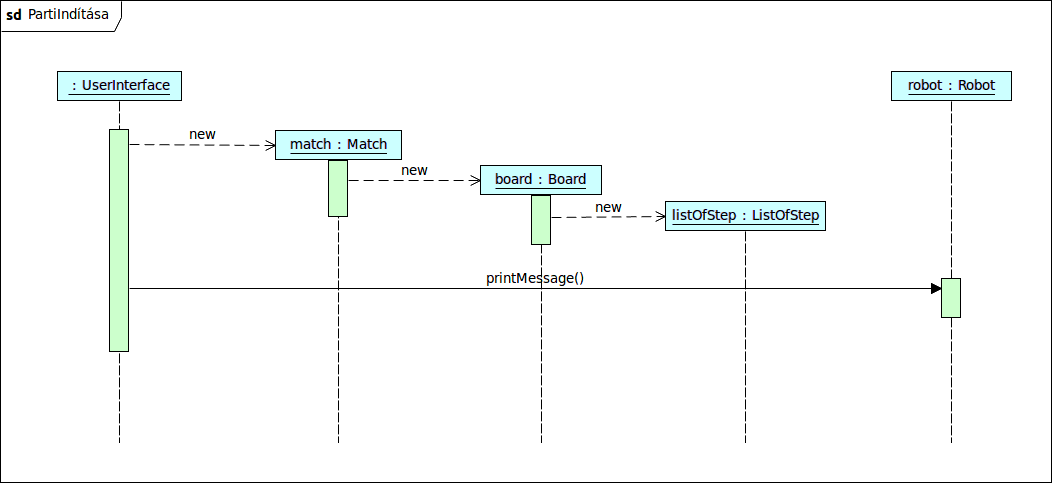
## Use Case PartiVége

UC07

# Diagram : PartiIndítása

From Interaction PartiIndításaDetail

)/]' maxW='true' keepH='false'>



## Message printMessage

Kiírja: "Parti idul."

# Diagram : DeploymentView

From Package DeploymentView

)/]' maxW='true' keepH='false'>

