**Potyogós**

Szoftvertechnológia házi feladat

Rittgasszer Ákos

Z8WK8D

# A feladat leírása

A játékot N játékos játssza (N >= 2, de tetszőlegesen nagy egész szám lehet). A játék egy N oldalú függőleges táblából áll, melybe forgatható tárcsák vannak beépítve. A tárcsák a tábla minden oldalán látszanak. Ha egy tárcsa forog, a tárcsa minden oldala együtt forog. A tárcsák szélén kis zsebek vannak. Egy tárcsa egyes oldalain a zsebek száma, illetve az zsebek pozíciója különböző is lehet.

Minden játékos néhány számozott koronggal (érmével) kezd a tábla tetején. Egy kis zsebbe pontosan egy korong illeszkedik. Ha két érintkező tárcsa zsebei egymás mellé kerülnek, a felső zsebben lévő korong átpottyan az alsó zsebbe.

A játék célja az érmék eljuttatása a tábla tetejéről a tábla alján lévő tálcába a tárcsák forgatásával. A játékosok felváltva forgatnak egy-egy tárcsát, de az a tárcsa nem mozgatható, amelyet az előző játékos éppen mozgatott. Amíg a játékos soron van, a kiválasztott tárcsát bármilyen pozícióba forgathatja, akár több irányba is és akár többször is. A győztes az a játékos, akinek először pottyan le az összes érméje a tálcába. A korongoknak a számozás sorrendjében kell leérkezniük.

Mivel egyik játékos sem látja a többi játékos tábláját, gyakran előfordulhat, hogy segítik vagy éppen hátráltatják a többiek előrehaladását. A játék arra késztet, hogy előre tervezzünk: belehajszolhatjuk a többi játékost, hogy a saját tárcsája forgatásával a mi korongjainkat vigye tovább, de közben vigyázzunk arra, nehogy rossz sorrendben potyogjanak le a korongjaink a tálcára.

# Funkcionális követelmények

## Elsődleges követelmények

|  |  |  |
| --- | --- | --- |
| **Azonosító** | **Leírás** | **Use-case** |
| R01 | A játék egy N oldalú függőleges táblából áll, melybe forgatható tárcsák vannak beépítve | View board, Move disc |
| R02 | A tárcsák a tábla minden oldalán látszanak | View board |
| R03 | Ha egy tárcsa forog, a tárcsa minden oldala együtt forog | Move disc |
| R04 | A tárcsákon zsebek vannak, számuk és helyzetük különböző | View board, Create game |
| R05 | Ha két érintkező tárcsa zsebei egymás mellé kerülnek, a felső zsebben lévő korong átpottyan az alsó zsebbe | Move balls |
| R06 | A játékosok felváltva forgatnak egy-egy tárcsát | Move disc |
| R07 | Az a tárcsa nem mozgatható, amelyet az előző játékos éppen mozgatott | Move disc |
| R08 | Amíg a játékos soron van, a kiválasztott tárcsát bármilyen pozícióba forgathatja, akár több irányba is és akár többször is | Move disc |
| R09 | A korong belép a játékba, bekerül a kiindulási tárolóból az első zsebbe | Move balls |
| R10 | A korong kilép a játékból, bekerül a végső tárolóba az utolsó zsebből | Move balls |
| R11 | Mindegyik táblán ugyanúgy vannak a zsebek | Creat game |

## További követelmények

|  |  |  |
| --- | --- | --- |
| **Azonosító** | **Leírás** | **Use-case** |
| R12 | Minden tárcsán van legalább egy zseb | Create game |
| R13 | Ha rossz számú korong esik le alura a játékos kiesik | Move balls, Lose game |
| R14 | Ha minden korióong lent van jó sorrendben a játékos nyer (egy nyertes van) | Move balls, Win game |

# Use-case-ek

## A close up of a map Description automatically generatedUse-case diagram

## Use-case leírások

|  |  |
| --- | --- |
| **Cím** | **Move disc** |
| **Leírás** | A játékos tudják a saját körükben mozgatni a korongot, a controller pedig a játékos által mozgatott koronggal szinkronban mozgatja a többi játékosét |
| **Aktorok** | Player, Controller |
| **Főforgatókönyv** | 1. A játékos mozgat egy kiválasztott korongot a köre végéig 2. A controller mozgatja a többi játékos korongját szinkronban 3. A korong zsebeibe kis és be tudnak esni a számozott labdák |
| **Alternatív forgatókönyv** | 1. Ha azt a korongot akarja mozgatni a játékos amit az előző körben mozgatott az előző játékos akkor az nem lehetséges |

|  |  |
| --- | --- |
| **Cím** | **View board** |
| **Leírás** | A játékos megnézi a saját tábláját (többiekét nem látja) |
| **Aktorok** | Player |
| **Főforgatókönyv** | 1. Kirajzolódik a tábla aktuális állapota 2. A játékos megnézi a saját táblája aktuális állapotát |

|  |  |
| --- | --- |
| **Cím** | **Move Balls** |
| **Leírás** | A labdák mozognak a korongok zsebei között |
| **Aktorok** | Controller |
| **Főforgatókönyv** | 1. Ha két zseb egymás mellé ér akkor a fentebbiből átesik a lentebbibe a labda |
| **Alternatív forgatókönyv** | 1. A legfelső korong zsebeibe egy tárolóból (amiben sorban vannak a labdák) tud beleesni 2. A legalsó korong zsebeiből kiesve a labda kiesik a játékból: 3. Ha a kieső labda száma ez előző kiesőnél eggyel nagyobb (és az 1-es labda esett ki először) akkor a játékos közelebb kerül a győzelemhez 4. Ha a kiső labda száma nem eggyel nagyobb az előtte kiesőénél, vagy nem az 1-es számú labda esik ki először, akkor a játékos vesztett |

|  |  |
| --- | --- |
| **Cím** | **Create game** |
| **Leírás** | A controller legenerálja a táblákat véletlenszerűen a játék elején |
| **Aktorok** | Controller |
| **Főforgatókönyv** | Legenerálja táblákat, mindegyik játékosnál ugyanúgy helyezkednek el a korongok, csak a zsebek helyzete eltérő |

|  |  |
| --- | --- |
| **Cím** | **Lose game** |
| **Leírás** | Veszít a játékos, nem a megfelelő korongja esik le |
| **Aktorok** | Player |
| **Főforgatókönyv** | Kiesik a játékból |

|  |  |
| --- | --- |
| **Cím** | **Win game** |
| **Leírás** | Nyer a játékos, lent van minden korongja |
| **Aktorok** | Player |
| **Főforgatókönyv** | Vége a játéknak, megvan a nyertes |

# Strukturális leírás

## Az osztályok leírása

### Game

**Felelősségek**

Végigviszi a játékot. Legenerálja a táblákat, aztán pedig addig mennek a körök amíg vége nem lesz a játéknak. A végén kiírja az eredményt.

**Attribútumok**

|  |  |
| --- | --- |
| -boards: Board[0…\*] | A játékhoz tartozó táblák |

**Metódusok**

|  |  |
| --- | --- |
| +Init() | A jéték kezdetén hívódik meg és legenerálja a játékosok számának megfelelő számú táblát. Megmondja mindegyik tárcsáról, hogy melyik tárcsákkal szomszédos (board.neighbourDiscs). Minden táblán ugyanúgy helyezkednek el a tárcsák, csak a zsebek elhelyezkedése eltérő |
| +Play(): void | A tényleges játékot halytja végre. Addig játszanak a játékosok a táblákon amíg nincsen vége a játéknak.  Eltárolja az előzőleg mozgatott tárcsát. Ellenőrzi az esetleges nyertest és a veszteseket |
| +End(winner: Player): void | Ha vége van a játéknak a Play() meghívja az End() függvényt. Ennek paramétere a nyertes játékos |

### Board

**Felelősségek**

A táblákon lehet végrehajtani az aktuális lépést. A lépés figyelembe veszi, hogy nem lehet az előzőleg forgatott korongot forgatni. A táblákhoz hozzá van rendelve a játékos akinek a tábla a játéktere. A táblához tartozik a felső és az alsó korongtároló is.

**Attribútumok**

|  |  |
| --- | --- |
| -gamer: Gamer | Egy táblához egy játékos tartozik. Ennek a játékosnak a körében lehet módosítani a táblát. |
| -upStorage: UpStorage | A labdák itt vannak alapból és innem tudnak bekerülni a játékba |
| -downStorage: DownStorage | Ide esnek le a labdák ha kiesnek az utolsó korongból. Ha kiesnek akkor a CheckLose() ellenőrzi, hogy jó-e a sorrend. Ha már minden labda kiesett, akkor a CheckWin() mondja meg, hogy jó sorrendben végzett-e |
| -discs: Disc[1…\*] | Ebben a tömbben vannak a táblához tartozó korongok. |

**Metódusok**

|  |  |
| --- | --- |
| +Step(movedDiscIdxÉ int): int | Egy lépés végrehajtásáért felelős. A ChoseDiskToMove függvény által kiválasztott tárcsára meghívja annak a Move metódusát. Visszatér az mozgatott korong indexével |
| +CheckWin(): bool | Visszatér egy logikai változóval, hogy van vagy nincs nyertes |
| +CheckLose(): bool | Visszatér, hogy vesztet-e az aktuális tábla játékosa |
| +RomoveCurrentBoard() | Eltávolítja az aktuális táblát a játékban levő táblák közül |
| -ChoseDiscToMove (movedDiscIdx: int): Disc | Kiválasztja az alapján, hogy mejik tárcsa volt mozgatva az előbb, hogy melyik less ebben a körben mozgatva és visszatér vele |

### Gamer

**Felelősségek**

Ez az osztály képviseli az egyes játékosokat amiket hozzárendelünk táblákhoz.

**Attribútumok**

|  |  |
| --- | --- |
| +name: string | Szöveg, ami a játékosok megkülönböztetésére alkalmas |

### Ball

**Felelősségek**

Ez az osztály képviseli a korongokat. Mindegyik korongnak avan egy száma, ami fontos a játékhoz. Ez az osztály kezeli azt, hogy a korongok átesnek az egyik helyről a másikba.

**Attribútumok**

|  |  |
| --- | --- |
| +number: int | A korongok száma, ami fontos a játékhoz |

### Disc

**Felelősségek**

Az eggyes tárcsák múködéséért felelős osztály

**Attribútumok**

|  |  |
| --- | --- |
| +neighbourDiscs: Disc[0...\*] | Egy tömb, amiben benne vannak a szomszédos tárcsák és a felső és also tárolók ha valamelyikükkel szomszédos |
| -pockets: Pocket[1…\*] | A tárcsához tartozó zsebek tömbje |
| -idx: int | A tárcsa indexe a táblán |

**Metódusok**

|  |  |
| --- | --- |
| +Move(): void | Forgatja a tárcsát, ha talál a szomszédos tárcsák (vagy a felső illetve also tároló) között olyan zsebet amiből át tud esni vagy amibe be tud esni a labda (ami a megfelelő zsebben van), akkor az átesik |
| +FallIn(neighbourDisc: Disc[0...\*]): BallStorage | Visszatér azzal a tárolóval ahonnan be tud esni az aktuális tárcsába korong |
| +FallIn(neighbourDisc: Disc[0...\*]): BallStorage | Visszatér azzal a tárolóval ahova be tud esni az aktuális tárcsából korong |

### BallStorage

**Felelősségek**

A korongok tárolására alkalmas abstract osztály, az új adstot a tömb végére teszi be

**Attribútumok**

|  |  |
| --- | --- |
| #numOfBalls: int | Az aktuálisan a tárolóban levő korongok száma |
| -balls: Ball[0…\*] | A tárolóban levő korongok tömbje |

**Metódusok**

|  |  |
| --- | --- |
| #AddBall(ball: Ball): void | Hozzád egy korongot a tárolóhoz, a végére rakja be |
| #RemoveBall(ball: Ball): void | Eltávolítja a paraméterül kapott elemet a tárolóból |
| +Fall(to: BallStorage): void | Az aktuális tárolóból atesik a korong ami át tud esni (ha van ilyen) a paraméterül kapottba |
| +GetBall(): Ball | Visszatér azzal a koronggal ami ki tud esni a tárolóból legközelebb, ha nincs ilyen akkor nullal |

### DownStorage

**Felelősségek**

A BallStorage osztályból származik le, működése megeggyezik a szülőjével. Azokat a korongokat tárolja amik kiestek a táblából.

### UpStorage

**Felelősségek**

A BallStorage osztályból származik le, működése megeggyezik a szülőjével. Azokat a korongokat tárolja amik még nincsenek játékba.

### Pocket

**Felelősségek**

A BallStorage osztályból származik le. Abban különbözik attól, hogy csupán egy korong tárolására alkalmas.

**Attribútumok**

|  |  |
| --- | --- |
| +deg: float | A tárcsán a függőlegessel bezárt szöge a zsebnek, forgatás során változik. Ezzel elhet megmondani, hogy át tud-e esni labda két zseb között |

**Metódusok**

|  |  |
| --- | --- |
| +Fall(to: BallStorage): void | Felülírja a szülője azonos nevű függvényét, úgy hogy cask olyan tárolóba eshet be labda ahol még nincsen |

## Osztálydiagram

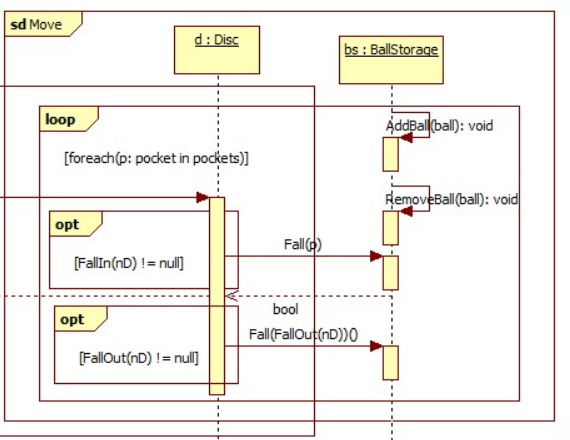
A close up of a device

Description automatically generated

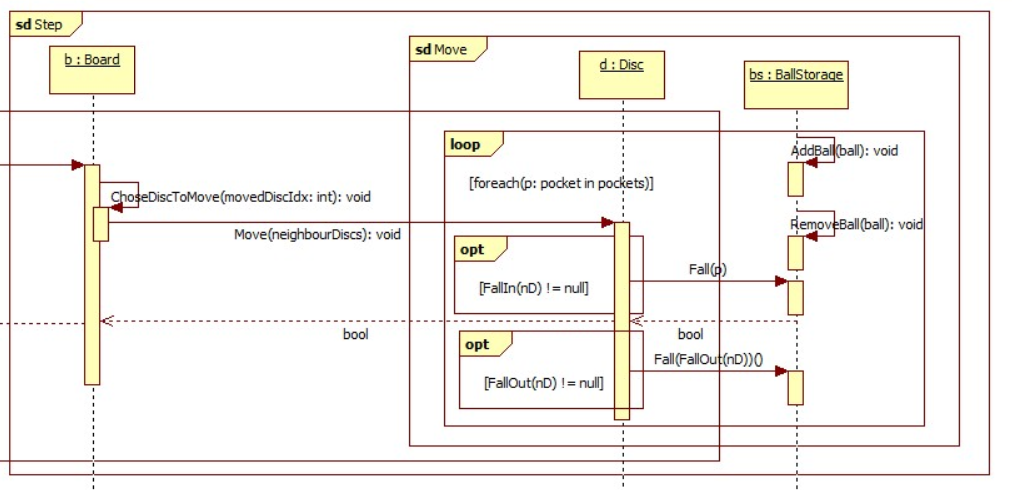
# Viselkedés leírása

## Szekvencia diagramok

### Move



### Step



### Play

### A screenshot of a social media post Description automatically generatedGame

# Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Elvégzett munka** | **Hivatkozások** |
| 2019.10.23. 15:00 | 1.5 óra | Követelmények felsorolása, use-case-ek összegyüjtése és a diagram elkészítése | 2, 3 |
| 2019.10.26. 11:00 | 4 óra | Osztályok leírása, osztálydiagramm elkészítése | 4 |
| 2019.11.03. 16:00 | 2 óra | Osztályok újragondolása, osztálydiagramm korrigálása | 4 |
| 2019.11.6. 20:00 | 4 óra | Szekvenciadiagramm megszerkesztése | 5 |
| 2019.11.9. 17:00 | 5 óra | Az egész újra átgondolása és kisebb módosítása | 2, 3, 4, 5 |
| 2019.11.10. 18:00 | 1 óra | Átnézés, kisebb hibák javítása. Dokumentáció véglegesítése | 2, 3, 4, 5 |

**Összes elvégzett munka:**17.5 óra

**Modellező eszköz:** WhiteStarUML

**Egyéb eszközök:** Microsoft Word