**8. Részletes tervek**

66 – otthonmeglefordult

Konzulens:

Szőke Máté

Csapattagok

|  |  |  |
| --- | --- | --- |
| Kovács Ákos | H7FTHG | [akoskovacs1980@gmail.com](mailto:akoskovacs1980@gmail.com) |
| Kaszala Kristóf | S9XEU5 | kaszalakristof1@gmail.com |
| Lőrincz Zoltán | BUXM3K | zoltan9811@gmail.com |
| Horesnyi Olivér | D7DBWE | horesnyi.oliver@freemail.hu |
| Hulej Attila | VTZ65K | alittahu@gmail.com |

2019.04.07.

Tartalomjegyzék

[8. Részletes tervek 3](#_Toc5549197)

[8.1 Osztályok és metódusok tervei. 4](#_Toc5549198)

[8.1.1 Animal 4](#_Toc5549199)

[8.1.2 Armchair 5](#_Toc5549200)

[8.1.3 ChocolateMachine 5](#_Toc5549201)

[8.1.4 ChocolatePanda 6](#_Toc5549202)

[8.1.5 Entrance 6](#_Toc5549203)

[8.1.6 Exit 7](#_Toc5549204)

[8.1.7 Field 7](#_Toc5549205)

[8.1.8 GamblerPanda 8](#_Toc5549206)

[8.1.9 Game 8](#_Toc5549207)

[8.1.10 Gamemachine 9](#_Toc5549208)

[8.1.11 ITickable 10](#_Toc5549209)

[8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén 10](#_Toc5549210)

[8.2.1 1. Panda step field 10](#_Toc5549211)

[8.2.2 2. Panda step WeakTile 13](#_Toc5549212)

[8.2.3 3. Panda falls in Hole 15](#_Toc5549213)

[8.2.4 4. Panda step on free Wardrobe 17](#_Toc5549214)

[8.2.5 5. Panda step on not-free Wardrobe 20](#_Toc5549215)

[8.2.6 A tesztelést támogató programok tervei 24](#_Toc5549216)

[8.3 Napló 25](#_Toc5549217)

# Részletes tervek

[A dokumentum célja, hogy pontosan specifikálja az implementálandó osztályokat, beleértve a privát attribútumokat és metódusokat, ezek definícióját is.

A dokumentum második fele részletesen be kell mutassa a korábban definiált be- és kimeneti nyelv szintakszisát felhasználva, hogy mely tesztekkel lesz a prototípus ellenőrizve.]

***8.0 Változások a bemeneti nyelvben***

A wardrobe létrehozása és kezelése kis mértékben megváltozott:

A létrehozáskor megadható a rajta álló állat, is ez akkor használható ha el szeretnék torlaszolni, ezáltal a párjáról nem lehet rá teleportálni.

Az új létrehozó parancs:

**Szekrény létrehozása**

create wardrobe [name] [wpair=null] [animal=null]

Leírás: Létrehoz egy szekrényt.

Opciók: name – A szekrény neve, később ezzel a névvel lehet rá hivatkozni.

Egyedinek kell lennie.

wpair – A szekrény párjának a neve, lehet null is ha még nem létezik,  
 de később beállítandó.

animal – A szekrényen álló állat neve, ha nem áll rajta senki akkor null.

Továbbá két szekrény összekötésére is kell egy új parancs. Eddig már a szerkény létrehozásakor meg kellett adni a párját, de ezt nem lehet megtenni mert az utasítások sorban jönnek és ekkor valamelyik még nem létezik. A megoldás az, hogy létre lehessen hozni egy szekrényt null értékű szomszéddal, és utána is be lehessen állítani.

**Szekrény párjának hozzáadása**

set w\_pair [wardrobe] [wpair\_to\_add]

Leírás: Beállítja egy szekrény párját.

Opciók: wardrobe – A szekrény neve, ehhez lesz beállítva a pár

wpair\_to\_add – A szekrény párjának a neve, amit be kívánunk állítani.

Ezek a változások az előző dokumentumba is át lettek vezetve, mivel azt használjuk a nyelv referenciájaként.

## Osztályok és metódusok tervei.

### Animal

#### Felelősség

Absztrakt osztály, az állatok belőle öröklődnek és valósítják meg az osztály egyes metódusait. Az állatok leírására szolgáló osztály.Egy általános állat, meg tud halni, lépni tud és más állatok nekimehetnek.

#### Ősosztályok

Nincs ősosztálya

#### Interfészek

Nem valósít meg interfészt.

#### Attribútumok

* **private Field field**: Pályát alkotó mező, amihez egy állatot tudunk rendelni, például Orángutánt. A mező amin az állat áll.
* **private double angle**: A szög amerre az állat néz, fokban. A 0 fok, ha az állat felfele néz.
* **private Game game**: Játék típusú változó. Itt tároljuk, hogy adott állat melyik játékban vesz részt. A játékot kezelő objektum.

#### Metódusok

* **public Field getField():** Megadja egy állatnak a tartózkodási helyét(mezőjét).
* **public void setField(Field field):** Beállítja egy adott állat tartózkodási helyét(mezőjét)
* **public double getAngle():** Megadja egy állatnak az irányát.
* **public void setAngle(double angle):** Beállítja egy állatnak az irányát
* **public Game getGame()** : Megadja egy adott állatról, hogy melyik játékban van.
* **public void setGame(Game game):** Beállítja, hogy az adott állat melyik játékban van
* **public void die()**: Az állat meghal és eltűnik a pályáról.
* **public void step(Field to)**: Az állat a megadott mezőre lép.
* **public void steppedOnField(Field f)**: Az állat csempére lépett.
* **public void steppedOnWeakTile(WeakTile wt)**: Az állat gyenge csempére lépett.
* **public void steppedOnChocolateMachine(ChocolateMachine cm)**: Az állat csokiautomatára lépett.
* **public void steppedOnArmchair(Armchair am)**: Az állat fotelra lépett.
* **public void steppedOnExit(Exit ex)**: Az állat kijáratra lépett.
* **public void steppedOnGameMachine(GameMachine gm)**: Az állat játékautomatára lépett.
* **public void steppedOnEntrance(Entrance ent)**: Az állat bejáratra lépett.
* **public void steppedOnWardrobe(Wardrobe w)**: Az állat szekrényre lépett.
* **public void HitByOrangutan(Orangutan o)**: Az állattal ütközik egy orángután.
* **public void HitByPanda(Panda p)**: Az állattal ütközik egy panda.
* **public void gotScared():** Az állat megijed.
* **public void jump():** Az állat ugrik.

### Armchair

#### Felelősség

A fotel leírására szolgáló osztály. A játékban a fotel lesz, egy mezőtípus, erre tud majd leülni a lusta panda. Egyszerre csak egy panda tud rajta ülni, amikor az kipihente magát, feláll a fotelből.

#### Ősosztályok

Field -> Armchair

#### Interfészek

Megvalósítja az ITickable interfészt.

#### Attribútumok

* **private Random rand:** Random változó, a fotelhez kötődő randomizáláshoz.

#### Metódusok

* **public Armchair():** Konstruktor
* **public void standUp():** Ha egy állat a fotelben ül, akkor egy random szomszédos mezőre lép miután feláll
* **public void tick():** a standUp() metódust hívja meg
* **public void wantToSitDown(LazyPanda lp):** Leülteti a lusta pandát a fotelbe
* **public void accept(Animal a):** Meghívja az állat steppedOnArmchair() metódusát

### ChocolateMachine

#### Felelősség

A csokiautomata leírására szolgáló osztály. A csokiautomata egy mezőtípus, a csokipandára van hatással. Időnként sípol egyet (bizonyos tick-ekre), ettől a szomszédos csokipandák megijednek.

#### Ősosztályok

Field -> ChocolateMachine

#### Interfészek

ITickable

#### Attribútumok

* **private boolean whistling:** Az értéke attól függően, hogy igaz vagy hamis, adja meg, hogy a gép éppen sípol-e, vagy sem.
* **private int tickCounter:** Egy számláló, aminek az értéke a gép sípolásához köthető
* **private Random random:** Random változó, csokigéphez kötődő randomizáláshoz

#### Metódusok

* **public boolean isWhistling():** Megadja, hogy a csokigép éppen sípol-e.
* **public void setWhistling(boolean whistling):** Beállítja a csokigépet, hogy sípoljon, vagy ne sípoljon, ezt a whistling attribútum igaz/hamis értékre való megváltoztatásával teszi.
* **public void getTickCounter():** Visszaadja a tickCounter értékét.
* **public void setTickCounter(int tickCounter):** Beállítja a tickCounter értékét
* **public ChocolateMachine():** Konstruktor
* **void tick():** Ha nem sípol az automata, véletlenszerűen elkezd sípolni, vagy nem. Ha sípol, ugrasztja a szomszédos mezőkön lévő állatokat és növeli a tickCountert. Ha a tickCounter eléri a 10-et, abbamarad a sípolás és nullázódik a számláló.
* **public void accept(Animal a):** Meghívja az adott állat steppedOnChocolateMachine() metódusát

### ChocolatePanda

#### Felelősség

A csokipanda leírására szolgáló osztály*.* Ha meghallja a csokiautomata sípolását, ugrik egyet.

#### Ősosztályok

Animal -> Panda -> ChocolatePanda

#### Interfészek

Nem valósít meg intefészt

#### Attribútumok

Nincs attribútuma

#### Metódusok

* **public ChocolatePanda():** Konstruktor
* **public void jump():** Meghívja az animalJumped() függvényét az állatnak. A csokipanda ugrik egyet.

### Entrance

#### Felelősség

A bejárat leírásáért felelős osztály. A bejárat egy mezőtípus. Ezen jön be az orángután, miután a kijáraton kiment.

#### Ősosztályok

Field -> Entrance

#### Interfészek

Nem valósít meg interfészt.

#### Attribútumok

Nincs attribútuma

#### Metódusok

* **public void enterToField(Orangutan o):** Elhelyezi az orángutánt az adott mezőn (az orángután fieldje az entrance obejktum lesz, az entrance animalja pedig az orángután).
* **public void accept(Animal a):** Meghívja az állat steppedOnEntrance() metúdusát.
* **public Entrance():** Konstruktor

### Exit

#### Felelősség

Az kijárat leírására szolgáló osztály. A kijárat egy mezőtípus, ezen kell kivinni az orángutánnak a pandákat. A kijáraton való áthaladáskor a játékos a pandák száma után pontot kap. A kivitt pandák időzítettségének megvonása az exit felelőssége

#### Ősosztályok

Field -> Exit

#### Interfészek

Nem valósít meg intefészt

#### Attribútumok

* **private Entrance entrancePair:** A kijárat ebben az attribútumban tárolja a hozzá tartozó bejáratot, ahol a kijáraton átlépő orángutánnak be kell majd jönnie.
* **private Game game:** A játékot kezelő objektum.

#### Metódusok

* **public Entrance getEntrance()**
* **public void setEntrance(Entrance entracePair)**
* **public Game getGame()**
* **public void setGame(Game game)**
* **public Exit():** Konstruktor
* **public void orangutanExited(Orangutan o):** Lekérdezi a kilépő orángután pandáinak listáját. A játékban az orángután pandáinak számával csökkenti az aktuális pandák számát a game-en keresztül. A pandák számának megfelelően pontozza az orángutánt irányító játékost a game-en keresztül. Törli a kivitt pandákat az orángután láncából továbbá leveszi őket a Timer listájáról és a hozzájuk tartozó mezőkről (null-ra állítja a mezők animal referenciáját). Meghívja az entrancePair enterToField() metódusát az átvett orángutánt paraméterként átadva.
* **public void accept(Animal a):** Meghívja az állat megfelelő steppedOn() metódusát, önmagát adva paraméterként.

### Field

#### Felelősség

Egy pályabeli csempét reprezentál és ősosztálya a különböző pályaelemeknek. Állatok léphetnek rá és el is tárolja, hogy mely állat tartózkodik rajta. Ismeri a szomszédos mezőket is.

#### Ősosztályok

Nincs ősosztálya.

#### Interfészek

Nem valósít meg intefészt.

#### Attribútumok

* **private List<Field> negihbours:** a mező szomszédai
* **private Animal a:** a mezőn tartózkodó állat. Ha nincs állat a mezőn, az értéke null.

#### Metódusok

* **public List<Field> getNeighbours()**
* **public void setNeighbours(List<Field> neighbours)**
* **pubic void addNeighbour(Field f)**
* **public Animal getAnimal()**
* **public void setAnimal(Animal a)**
* **public Field():** Konstruktor
* **public Field getNeighbourByAngle(double angle):** Visszaadja a szomszédjai közül azt a mezőt, amely a paraméterben átadott szöghöz tartozik
* **public void accept(Animal a):** Fogadja a paraméterben átadott állatot. Hívja az állat megfelelő (a mező típusától függő) metódusát.
* **public void animalJumped(Animal a):** Egy állat hívja meg, aki ugrott az adott mezőn.
* **public void wantToSitDown(LazyPanda lp):** Egy LazyPanda hívja meg, aki le akar ülni az adott mezőn.

### GamblerPanda

#### Felelősség

A játékos pandát leíró osztály. Ha meghallja a játékgép csilingelését, megijed.

#### Ősosztályok

Animal -> Panda -> GamblerPanda

#### Interfészek

ITickable

#### Attribútumok

Nincs attribútuma

#### Metódusok

* **public GamblerPanda():** Konstruktor
* **public void gotScared():** A panda megijed, emiatt ha egy láncba tartozik, akkor elengedi a többi panda kezét (removePandaFromLine() metódus).

### Game

#### Felelősség

A játékot leíró osztály. Elindítja és leállítja a játékot. Kezeli a játékosokat és a pontjaikat, illetve a pandák számát

#### Ősosztályok

Nincs ősosztálya

#### Interfészek

Nem valósít meg interfészt.

#### Attribútumok

* **private int pandaNum:** A pályán lévő pandák aktuális száma
* **private Timer timer:** A játék időzítő objektuma.
* **private int timeFromStart:** A játék indulása óta eltelt idő.
* **private List<Player> players:** A játékosokat tárolja

#### Metódusok

* **public Game():** Konstruktor
* **public Timer getTimer():** Visszaadja a timer-t
* **public void setTimer(Timer t):** Beállítja a timer értékét
* **public void setPandaNum(int num):** Beállítja a pandaNum értékét
* **public void start()**: Elindítja a játékot.
* **public void endGame()**: Leállítja a játékot.
* **public void decreasePandaNum(int i)**: Csökkenti az aktuális pandák számát i-vel.
* **public void increaseTime()**: Növeli az eltelt időt eggyel.
* **public void addPoints(int p, Orangutan o)**: Az adott orángutánhoz tartozó játékosnak p pontot ad.

### Gamemachine

#### Felelősség

A játékgépet leíró osztály. A játékgép egy mezőtípus, ami néha csilingel (bizonyos tick-ekre), ezzel hatással van a környező játékfüggő pandákra.

#### Ősosztályok

Field -> Gamemachine

#### Interfészek

Nem valósít meg interfészt.

#### Attribútumok

* **private boolean ringing:** Attól függően igaz vagy hamis az értéke, hogy csilingel-e az automata.
* **private int tickCounter:** A csilingelés kezdete óta eltelt idő.
* **private Random random:** Random változó, játékgéphez kötődő randomizáláshoz

#### Metódusok

* **public boolean isRinging():** Visszatérési értéke igaz vagy hamis, attól függően, hogy a gép éppen csilingel vagy sem.
* **public void setRinging(boolean ringing):** Beállítja, hogy a gép csilingeljen, vagy ne csilingeljen, attól függően, hogy igaz vagy hamis értéket adunk meg neki.
* **public int getTickCounter():** Visszaadja a csilingelés kezdete óta eltelt időt
* **public void setTickCounter(int tickCounter):** Beállítja a tickCounter értékét.
* **public GameMachine():** Konstruktor
* **public void accept(Animal a):** A játékgép nem fogad egyetlen állatot sem, a metódus azonnal visszatér.
* **public void tick():** Ha nem csilingel az automata, véletlenszerűen elkezd csilingelni, vagy nem. Ha csilingel, ugrasztja a szomszédos mezőkön lévő állatokat és növeli a tickCountert. Ha a tickCounter eléri a 10-et, abbamarad a csilingelés és nullázódik a számláló.

### ITickable

#### Felelősség

Az ITickable egy interfész, amelyet minden olyan osztály megvalósít, amelynek bizonyos (véletlenszerű vagy meghatározott) időközönként kell csinálnia valamit.

#### Ősosztályok

Nincs ősosztálya

#### Interfészek

Az ITickable maga egy interfész

#### Attribútumok

Nincs attribútuma

#### Metódusok

* **public void tick():** A bizonyos időközönként elvégzendő dolgok

## A tesztek részletes tervei, leírásuk a teszt nyelvén

[A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokat, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel. A tesztek leírásakor az előző dokumentumban (proto koncepciója) megadott szintakszist kell használni.]

### 1. Panda step field

* **Leírás**

Azon folyamat modellje, amikor panda egy szomszédos, üres csempére lép.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A panda lépését teszteljük, a teszt során azt várjuk, hogy a lépés sikerül, mert a cél csempe üres.

* **Bemenet – test\_01\_in.txt**

# 1. Panda step Field

# Rövid leírás

# Azon folyamat modellje, amikor panda egy szomszédos, üres #csempére lép.

# Teszt célja

# A panda rálép egy sima Field-re.

# +---------+

# | f1 - p1 |

# +---------+

# | f2 |

# +---------+

#

# |

# |

# |

# \|/

# ˇ

#

# +---------+

# | f1 |

# +---------+

# | f2 - p1 |

# +---------+

random off

create timer t1

create game g1 1 t1

create panda p1 g1 null null

add tickable t1 p1

create field f1 p1

set a\_field p1 f1

create field f2 null

add neighbour f1 f2

add neighbour f2 f1

step p1 f2

save test1\_out.txt

* **Elvárt kimenet – test\_01\_exout.txt**

Timer:

t1

t: p1

Game:

g1

pn: 1

t: t1

tfs:

ps:

Panda:

p1

g: g1

l: null

f: f2

Field:

f1

a: null

n: f2

Field:

f2

a: p1

n: f1

### 2. Panda step WeakTile

* **Leírás**

Azon folyamat modellje, amikor a panda egy szomszédos gyenge csempére lép, aminek az életereje nem 0.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A panda lépése, és a gyenge csempe életének változását teszteljük. Azt várjuk hogy a lépés sikeres, mert szabad a gyenge csempe, és a gyenge csempe élete 1-gyel csökken.

* **Bemenet – test\_02\_in.txt**

# 2. Panda step WeakTile

# Rövid leírás

# Azon folyamat modellje, amikor a panda egy szomszédos gyenge #csempére lép, aminek az életereje nem 0.

# Teszt célja

# A panda rálép egy gyenge csempére, aminek ennek hatására #csökken az életereje.

# +---------+

# | f1 - p1 |

# +---------+

# | wt1 |

# +---------+

# |

# |

# |

# \|/

# ˇ

# +---------+

# | f1 |

# +---------+

# | wt1- p1 |

# +---------+

random off

create timer t1

create game g1 1 t1

create panda p1 g1 null null

add tickable t1 p1

create field f1 p1

set a\_field p1 f1

create weaktile wt1 null 20

add neighbour f1 wt1

add neighbour wt1 f1

step p1 wt1

save test2\_out.txt

* **Elvárt kimenet – test\_02\_exout.txt**

Timer:

t1

t: p1

Game:

g1

pn: 1

t: t1

tfs:

ps:

Panda:

p1

g: g1

l: null

f: wt1

Field:

f1

a: null

n: wt1

Waktile:

wt1

a: p1

l: 19

n: f1

### 3. Panda falls in Hole

* **Leírás**

A panda egy szomszédos gyenge csempére lép, aminek az életereje 0.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

A csempe törött, ezért a panda meghal. Az elvárt működés szerint kikerül a Timer osztály tickable listájából. A játékban a pandák száma 0 lesz, a panda mezője és az összes mező állata null.

* **Bemenet – test\_03\_in.txt**

**# 3. Panda falls in Hole**

**# Rövid leírás**

**# A panda egy szomszédos gyenge csempére lép, aminek az #életereje 0.**

**# Teszt célja**

**# A panda rálép egy törött gyenge csempére, aminek hatására meghal.**

**# +---------+**

**# | f1 - p1 |**

**# +---------+**

**# | wt1 |**

**# +---------+**

**# |**

**# |**

**# \|/**

**# ˇ**

**# +---------+**

**# | f1 |**

**# +---------+**

**# | wt1- p1 | p1 dies :(**

**# +---------+**

**random off**

**create timer t1**

**create game g1 1 t1**

**create panda p1 g1 null null**

**add tickable t1 p1**

**create field f1 p1**

**set a\_field p1 f1**

**create weaktile wt1 null 0**

**add neighbour f1 wt1**

**add neighbour wt1 f1**

**step p1 wt1**

**save test3\_out.txt**

* **Elvárt kimenet – test\_03\_exout.txt**

Timer:

t1

t:

Game:

g1

pn: 0

t: t1

tfs:

ps:

Panda:

p1

g: g1

l: null

f: null

Field:

f1

a: null

n: wt1

Weaktile:

wt1

a: null

l: 0

n: f1

### 4. Panda step on free Wardrobe

* **Leírás**

A panda egy szomszédos szekrény típusú mezőre lép, aminek a párja szabad.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Az elvárt működés az, hogy a panda kijön a szekrény párján, mivel a szekrény párja szabad.

* **Bemenet – test\_04\_in.txt**

# 4. Panda step on free Wardrobe

# Rövid leírás

# A panda egy szomszédos szekrény típusú mezőre lép, aminek a #párja szabad.

# Teszt célja

# A panda rálép egy szekrényre, aminek a párján jön ki.

# +---------+

# | f1 - p1 |

# +---------+

# | wd1 |

# +---------+

# | f2 |

# +---------+

# | f3 |

# +---------+

# | wd2 |

# +---------+

# | f4 |

# +---------+

#

# |

# |

# |

# \|/

# ˇ

# +---------+

# | f1 |

# +---------+

# | wd1 |

# +---------+

# | f2 |

# +---------+

# | f3 |

# +---------+

# | wd2- p1 |

# +---------+

# | f4 |

# +---------+

random off

create timer t1

create game g1 1 t1

create panda p1 g1 null null

add tickable t1 p1

# fields

create field f1 p1

set a\_field p1 f1

create wardrobe wd1 null null

create field f2 null

create field f3 null

create wardrobe wd2 wd1 null

set w\_pair wd1 wd2

create field f4 null

add neighbour f1 wd1

add neighbour wd1 f1

add neighbour wd1 f2

add neighbour f2 wd1

add neighbour f2 f3

add neighbour f3 f2

add neighbour f3 wd2

add neighbour wd2 f3

add neighbour wd2 f4

add neighbour f4 wd2

step p1 wd1

save test4\_out.txt

* **Elvárt kimenet – test\_04\_exout.txt**

Timer:

t1

t: p1

Game:

g1

pn: 1

t: t1

tfs:

ps:

Panda:

p1

g: g1

l: null

f: wd2

Field:

f1

a: null

n: wd1

Wardrobe:

wd1

wp: wd2

a: null

n: f1 f2

Field:

f2

a: null

n: wd1 f3

Field:

f3

a: null

n: f2 wd2

Wardrobe:

wd2

wp: wd1

a: p1

n: f3 f4

Field:

f4

a: null

n: wd2

### 5. Panda step on not-free Wardrobe

* **Leírás**

A panda egy szomszédos szekrény típusú mezőre lép, aminek a párja foglalt.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Az elvárt működés az, hogy a panda a helyén marad, mivel a szekrény párja foglalt.

* **Bemenet – test\_05\_in.txt**

# 5. Panda step on not-free Wardrobe

# Rövid leírás

# A panda egy szomszédos szekrény típusú mezőre lép, aminek a párja foglalt.

# Teszt célja

# A panda rálép egy szekrényre, de a teleport meghiúsul.

# +---------+

# | f1 - p1 |

# +---------+

# | wd1 |

# +---------+

# | f2 |

# +---------+

# | f3 |

# +---------+

# | wd2-p2 |

# +---------+

# | f4 |

# +---------+

#

# |

# |

# |

# \|/

# ˇ

# +---------+

# | f1 - p1 |

# +---------+

# | wd1 |

# +---------+

# | f2 |

# +---------+

# | f3 |

# +---------+

# | wd2- p2 |

# +---------+

# | f4 |

# +---------+

random off

create timer t1

create game g1 1 t1

create panda p1 g1 null null

create panda p2 g1 null null

add tickable t1 p1

add tickable t1 p2

# fields

create field f1 p1

set a\_field p1 f1

create wardrobe wd1 null null

create field f2 null

create field f3 null

create wardrobe wd2 wd1 p2

set w\_pair wd1 wd2

set a\_field p2 wd2

create field f4 null

add neighbour f1 wd1

add neighbour wd1 f1

add neighbour wd1 f2

add neighbour f2 wd1

add neighbour f2 f3

add neighbour f3 f2

add neighbour f3 wd2

add neighbour wd2 f3

add neighbour wd2 f4

add neighbour f4 wd2

step p1 wd1

save test5\_out.txt

* **Elvárt kimenet – test\_05\_exout.txt**

Timer:

t1

t: p1

Game:

g1

pn: 1

t: t1

tfs:

ps:

Panda:

p1

g: g1

l: null

f: f1

Panda:

p2

g: g1

l: null

f: wd2

Field:

f1

a: p1

n: wd1

Wardrobe:

wd1

wp: wd2

a: null

n: f1 f2

Field:

f2

a: null

n: wd1 f3

Field:

f3

a: null

n: f2 wd2

Wardrobe:

wd2

wp: wd1

a: p2

n: f3 f4

Field:

f4

a: null

n: wd2

### A tesztelést támogató programok tervei

[A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.]

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2019.04.04. –  2019.04.05.  23:00 – 01:00 | 2 óra | Kovács | Teszt 1-5 megírása, dokumentumba beillesztés, változások átvezetése. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |