

SZAKDOLGOZAT



MISKOLCI EGYETEM

Multiágens rendszerek koordinációs problémáinak vizsgálata

Készítette:

Mengyán Márton

Programtervező informatikus

Témavezető:

Piller Imre

MISKOLC, 2022

MISKOLCI EGYETEM

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

Szám:

SZAKDOLGOZAT FELADAT

Mengyán Márton (BPWTW0) programtervező informatikus jelölt részére.

A szakdolgozat tárgyköre: kulcsszavak, hasonló

A szakdolgozat címe: Multiágens rendszerek koordinációs problémáinak vizsgálata

A feladat részletezése:

A multiágens rendszerek célja, hogy egy adott problémát egyidejűleg több ágens irányításával tudjanak megoldani.

A dolgozat azt vizsgálja, hogy ilyen esetekben az ágensek között milyen jellegű kommunikáció lehet, illetve hogy az adott ágensnek hogyan kell viselkednie a környezetből, és más ágensektől érkező információk függvényében.

Ehhez specifikálásra, megtervezésre és implementálásra kerül egy szimulációs környezet, amely valós időben követhetővé teszi az ágensek állapotát.

Az ágensek viselkedéséhez tartozó paraméterek optimalizálásra kerülnek annak érdekében, hogy az adott célt minél hatékonyabban (például gyorsabban vagy kevesebb művelet elvégzésével) el tudják érni.

Témavezető: Piller Imre (beosztása)

A feladat kiadásának ideje: 2021 Szeptember 27.

.....
szakfelelős

EREDETISÉGI NYILATKOZAT

Alulírott **Mengyán Márton**; Neptun-kód: BPWTW0 a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős Programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy *Multiágens rendszerek koordinációs problémáinak vizsgálata* című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, év hó nap

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve:

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata:

a bíráló javaslata:

a szakdolgozat végleges eredménye:

Miskolc,

.....

a Záróvizsga Bizottság Elnöke

Tartalomjegyzék

1. Bevezetés	1
2. Konceptió	2
2.1. A fejezet célja	2
2.2. Felhasznált technológiák	2
2.3. Menü	4
2.4. UI	4
2.5. Szimulációs környezet specifikálása	6
2.5.1. Map	6
2.5.2. Szobák és folyosók	6
2.5.3. Map	7
2.6. Tárgyak	7
2.6.1. name	8
2.6.2. typeName	8
2.6.3. type	8
2.6.4. str	8
2.6.5. vit	8
2.6.6. eva	8
2.6.7. acc	8
2.7. Tárgy nevek	8
2.7.1. Entity Helmet 1	8
2.7.2. Entity Helmet 2	9
2.7.3. Entity Armor 1	9
2.7.4. Entity Armor 2	9
2.7.5. Entity Weapon 1	9
2.7.6. Entity Weapon 2	10
2.7.7. Kulcs	10
2.8. Ágensek	10
2.8.1. Statisztika	10
2.8.2. Statisztikák	10
2.8.3. Statisztikák jelentősége	11
2.8.4. Sebzés számlálás	11
2.9. Ágens főbb tulajdonságai	11
2.9.1. Kinézetet befolyásoló tényező	12
2.9.2. Kommunikáció közöttük	12
2.9.3. Megfigyelési akcióik	12
2.9.4. Ágens tevékenységeinek a köre	13
2.9.5. Ágens fő céljai	14

2.9.6.	Ágens másodlagos céljai	14
2.9.7.	Ágens függvény	14
2.10.	Cél Tárgy	16
2.11.	Combat	16
2.12.	Interface felület	16
2.12.1.	Pause	16
2.12.2.	Inventory	16
2.12.3.	Karakter	16
2.13.	Mapon található mozgást kizáró objektumok	17
2.13.1.	Brick	17
2.13.2.	Zárt ajtó	17
2.14.	Idő	17
2.15.	Kamera	17
2.16.	Irányítás	17
3.	Tervezés	24
3.1.	Adatmodellek	24
4.	Megvalósítás	26
4.1.	wantToAttack	27
4.2.	wantToEquip	27
4.3.	wantToPickUp	27
4.4.	wantToDeleteAnItem	27
4.5.	wantToOpenDoor	28
4.6.	wantToMoveToItem	28
4.7.	wantToMove	28
5.	Tesztelés	30
6.	Összefoglalás	31
	Irodalomjegyzék	32

1. fejezet

Bevezetés

A szakdolgozat egy szimulációs környezet létrehozásával fog kezdődni, amelyet felül nézetből vehetünk szemügyre. Körökre osztott, a karakterünket és az ágenseket egy fix sorrendbe rakjuk, amely meghatározza, mikor jön melyik karakter. A szimulációs környezet véges számú különböző szobákból és folyosókból épül fel. Két szobát egy folyosó köt össze, amelyet a folyosó mindkét végén egy ajtó zár le. Több részre van bontva, amelyek nem elérhetőek egyidejűleg.(?hány szint, 3-5?) Ezeket a részeket 'szint'-eknek nevezzük(?fix pályák vagy valamennyire randomizált pályák?). Minden szint különböző terep nehézségeket okoz. Pl. lassabb mozgás. Ahhoz hogy a következő szintre léphessünk tovább, teljesíteni kell az adott szintet először. Szintek közötti mozgás egy adott helyen történik, egy szobában, egy nem mozdítható objektum segítségével. Mivel a szimulációs környezet úgy mond 'dungeon'-t imitál, ezért a szintek közötti lépés egy lépcső használatával történik. Minden szinten adott, hogy milyen ágens veszt részt a szimulációban.

Az ágens két nagyobb csoportra vannak osztva, amelyeket 'frakció'-knak nevezünk. Minden frakciónak van egy teljesítendő célja, amelyet meg kell gátolnunk a továbbjutáshoz. A játékos is rendelkezik egy adott céllal, amelyet a frakciók szintén próbálnak meggátolni. A frakciók szabotálják a másik frakció céljainak elérését is. Ha a valamelyik frakció teljesíti a célját, vagy a játékos 'HP'-ja, másnéven életereje elfogy, akkor a játékos vereségével fejeződik be a játék.

A szobák és folyosók blokkokból épülnek fel. Ezeken a blokkokon helyezkedhetnek el a karakterek limitáltan, azaz egynél több karakter nem tartózkodhat rajta.

2. fejezet

Koncepció

2.1. A fejezet célja

A szimulációs környezet inspirációt vesz már elkészített szimulációokból, mint például a Pixel Dungeon és Darkest Dungeon nevű játékokból ismerhető szoba és folyosó kapcsolat. Ahol minden szobát egy folyosó köt össze egy másik tetszőleges szobával. A Darkest Dungeon-hoz képest a játékmenet rugalmasabb, mivel a szobák bármikor elhagyhatóak és megközelíthetőek, ha engedi azt a pálya jelenlegi belső felépítése. Azaz, ha nincs útban valami, ami megakadályozza.

Illetve inspirációt vesz még továbbá a For The King nevezetű játékból, ahol a fegyverek adott karakter statisztikákból "scalelődnek", azaz a birtokolt statisztikák függvényében számolódnak ki az adott értékek figyelembe véve a célpont általt birtokolt statisztikákat is.

A fent említett három játék mind körökre osztott, ahogyan a szimulációs környezetünk is az lesz.

A szimulációs környezetünk viszont egyedi lesz abból a szempontból, hogy az ágens csoportokra vannak osztva és közös céllal rendelkezve próbálják legyőzni egymást és a ágenst. Illetve hogy nem csak a ágensnek létezik 'win condition'-je, hanem az ágenseknek is, mindkét frakciónak különböző.

2.2. Felhasznált technológiák

Pixel Dungeon:

- Körökre osztott tevékenységekör.
- Ahogyan a referenciaként szolgáló játékképeken látszódik, a játék szobák és folyosók véges kombinációjából épül fel.
- A játéktér teljes része négyzet cellákból épül fel, amelyek összességét.
- A cellák tartalmazhatnak objektumokat és ágenseket, illetve tárgyakat.
- Illetve létezik a szint fogalma, de csak a játékot irányító számára.
- Tárgyak használata.
- Randomizált map kialakítás.

For The King

- Körökre osztott, a jövőbeli köröket láthatjuk dinamikusán a képernyőn a combat phase-ban.
- Statisztikák által behatárolt számolások.
- Tárgyak használata.
- Illetve létezik a szint fogalma, de csak a játékot irányító számára.
- Egyszerre több karakter irányítása.
- Részben tartalmaz szobákat, de azok egyirányúak.
- A játéktér bizonyos részei hexagon cellákból épül fel.

Darkest Dungeon

- Szobák és folyosók összessége, de több kisebb mapból épül fel.
- Randomizált map kialakítás.
- Tárgyak használata.
- Egyszerre több karakter irányítása.
- Körökre osztott, de nem láthatjuk a képernyőn a sorrendet.
- Illetve létezik a szint fogalma, de csak a játékot irányító számára.

A szimulációs környezet tartalmazni fogja a:

- Szobák és folyosók által felépített map struktúráját.
- A körök dinamikus kiszámolását és megjelenítését a szimulációban.
- Négyzet cellákból felépített szimulációstér.
- Tárgyakat és azok használatát.
- Szint fogalmát, de minden ágens hozzáfér.
- Egyszerre több karakter irányítását azonos célok által vezérelve.
- Statisztikák által meghatározott számolásokat.
- Randomizált mapokat.

2.3. Menü

A szimuláció indítása előtti ablak.

- Közép fent a Main Menu kiírását láthatjuk, allatta pedig 2 opcióból választhatunk:
- (1) Start Game, a kiválasztása esetén elindul a szimuláció.
- (2) Quit, a szimulációs ablak bezárása.
- A fenn említett két menü pont között a W(fel) és S(le) billentyűk lenyomásával navigálhatunk.
- Aláhúzva jelzi a felhasználónak, hogy mely menüpont van éppen kiválasztva.

A szimuláció megállítása esetén.

- A szimulációban a P billentyű lenyomásával, képes a felhasználó megállítani a szimuláció menetét.
- A megállítást a bal felső sarokban kiírt fehér Paused kiírás jelzi a felhasználó számára.
- Ilyenkor az ágensek nem kapnak lehetőséget a lépésre. De a felhasználó képes a kamera mozgatására a nyilak segítségével.
- A felhasználó szintén képes az ágensek statisztikája ablakot megnyitni/bezárni a C billentyű megnyomásával, és előre/hátra lapozni az ágensek között a Q(balra) és E(jobbra) billentyűk megnyomásával, ha több van a szimulációban, mint 1.
- Ugyanígy képes megnyitni/bezárni az adott ágenshez tartozó Inventory ablakot az I billentyű megnyomásával, ha látszódik az ágens statisztikája ablak.
- A P billentyű újonnan megnyomásával, ott folytatódik a szimuláció, ahol abba-maradt.

2.4. UI

- A szimuláció figyelésére szolgáló UI ablakok:
 - A szimuláció kezdete után a képernyőn megjeleníthető ablakok a következők:
 - Statistics: A C billentyű megnyomására megjelenik az első számú ágens Statistics ablaka.
 - Inventory: Az I billentyű megnyomására megjelenik az az adott Ágens Inventory oldala.
- Statistics
 - Egy adott ágens adatának a kiolvasásának az eredményeit megjelenítő UI ablak.

- A szimulációban, megállítástól függetlenül a C billentyű megnyomására jelenik meg, és újonnan megnyomására tűnik el.
 - Megnyitásakor mindig a még szimulációban létező ágensek közül az első számú ágens adatait fogja visszaadni.
 - A vizsgálandó Ágens elpusztulásakor a még létező ágensek közül az új első számú ágens adatait fogja visszatenni.
 - Ezen az ablakon a következő Ágens adatokat írja ki:
 - Sorszám: Az ágens sorszámát adja vissza.
 - HP: Az ágens aktuális életerejét adja vissza.
 - Str: Az ágens aktuális str értékét adja vissza.
 - Vit: Az ágens aktuális vit értékét adja vissza.
 - Eva: Az ágens aktuális eva értékét adja vissza.
 - Acc: Az ágens aktuális acc értékét adja vissza.
 - Ezek a számok változhatnak az alapján, hogy milyen felszerelést hord az adott ágens.
 - Új tárgy felvételénél, ha úgy ítéli, hogy jobb mint az általa hordott azonos típusú tárgy, akkor lecseréli, és ennek megfelelően frissülnek a statisztikái.
- Inventory
 - Nem nyitható meg, ha nincs még megnyitva egy tetszőleges ágens Statistics ablaka.
 - Megnyitásakor annak az ágensnek az Inventoryja lesz látható, amelyiknek a Statistics oldala nyitva van.
 - A Statistics oldalon van előre/hátra lépés esetén az Inventory az újonnan szemügyre vett ágens Inventoryját fogja megjeleníteni a felhasználó számára.
 - Az Inventoryban a következő típusú tárgyak találhatók meg:
 - Armor: Armorként hordható tárgy, 4 alapstatisztikát tartalmaz.
 - Helmet: Helmetként hordható tárgy, 4 alapstatisztikát tartalmaz.
 - Fegyver: Fegyverként hordható tárgy, 4 alapstatisztikát tartalmaz.
 - Key: Ajtó kinyitásához szükséges tárgy.
 - Az Inventoryban a WASD billentyűk lenyomásával választhatjuk ki, hogy mely Inventory helyet akarjuk megnézni.
 - A fehér üres négyzet, jelöli azt a helyet az Inventoryban, amelyet éppen szemügyre vesz a felhasználó.
 - Ha felszerelés típusú az adott tárgy, akkor a nevét és a 4 alapstatisztika értékét írja ki.
 - Ha nem felszerelés típusú az adott tárgy, akkor pedig a nevét írja ki.

2.5. Szimulációs környezet specifikálása

2.5.1. Map

Szimuláció kezdetekor megtörténik a map beolvasása:

- Az 50x50-es mátrixon minden cellának egy block típus beolvasása:
 - 0 típus: Üresség, a mapon nem elérhető területeket jelöli.
 - 1 típus: Fal, a bejárható területeket körbezáró fal, amelyen nem lehet átmenni.
 - 2 típus: Füves terület, az ágensok által bejárható területek.
- Az 50x50-es mátrixon minden 2-es típusú blockra helyezhetünk el egy objektumot.
 - Brick: Egy mozgást korlátozó objektum. Nincs különösebb szerepe.
 - Key: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - EntityArmor01: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - EntityWeapon01: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - EntityHelmet01: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - EntityArmor02: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - EntityWeapon02: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - EntityHelmet02: Mozgást nem korlátozó tárgy, amelyet az Ágensok képesek felvenni az Inventoryjukba.
 - Chest: End screen előidéző tárgy.
 - Door: Egy mozgást korlátozó objektum. Kulccsal kinyitható az Ágensok által.
- Az 50x50-es mátrixon minden 2-es típusú blockra helyezhetünk el egy ágens.
 - EntityFaction0: 0-es számú csapatban lévő Ágens.
 - EntityFaction1: 1-es számú csapatban lévő Ágens.

2.5.2. Szobák és folyosók

A szobákat több szomszédos cellák alkotják, ezeket a szobákat folyosók kötik össze, amelyeket pontosan kettő cellával szomszédos cellák alkotnak. Minden szomszédos szobát és folyosót egy ajtó választ el egymástól. Celláin korlátozottan helyezkedhetnek el térbeli objektumok (Lásd: oszlop), amely a cellát nem bejárható cellává alakít és korlátozza a látást.

2.5.3. Map

A map szobák és folyosók összesége, ahol a szobák alakjánál törekszünk a négyzet alakú szobák elkerülésére. A folyosók általában rövidek és egy szobából akár több is nyílik egy adott szobára. Minden maphoz tartoznak adatok.

Ezek az adatok a blockok típusát fejezi ki, amelyek $]0,2[$ nyílt intervallumon belüli számok jelölnek. Ezek az adatok a szimuláció kezdetekor kerülnek beolvasásra a world.txt-ből.

A mapon bejárható területeken szobák és folyosók összesége, ahol minden szobának és folyosóhoz tartozik adat, ez az adat a szobaID és folyosóID.

A szobákban és a folyosókban bejárható terület egy mátrixhoz hasonló, ahol a celláit blokkoknak nevezzük. Minden blokkhoz tartoznak adatok.

Ezek az adatok:

- X koordináta (Integer),
- Y koordináta (Integer),
- Block típusa (Integer).

Fontosabb kizáró feltételek:

- Ajtó csak szoba és a folyosó, vagy szoba és a szoba között létezzen.
- Folyosóból ne nyíljon ajtó folyosóra.
- Két különböző folyosó ne érjen össze.
- A map bármelyik részéből elérhető legyen a map bármelyik tetszőleges helye.
- Nem lehet olyan cellára mozgást kizáró objektumokat letenni, amely lehetetlenné teszi a map egy tetszőleges pontjáról a map egy másik tetszőleges pontjára való eljutását.

2.6. Tárgyak

A tárgyakhoz a következő fontosabb adatok tartoznak:

- name (String)
- typeName (String)
- type (Integer)
- str (Integer)
- vit (Integer)
- eva (Integer)
- acc (Integer)

2.6.1. name

Az adott tárgyat bemutató név.

2.6.2. typeName

A tárgyak tartalmazznak tárgynevet, lásd: Armor, Helmet, key... stb.

2.6.3. type

A Key tárgy a "0" típusú, amíg minden hordható tárgy a "2" típus számot tartalmazza.

2.6.4. str

Ha az adott tárgy az ágensek által hordható tárgy, azaz nem kulcs, van valós str értéke.

2.6.5. vit

Ha az adott tárgy az ágensek által hordható tárgy, azaz nem kulcs, van valós vit értéke.

2.6.6. eva

Ha az adott tárgy az ágensek által hordható tárgy, azaz nem kulcs, van valós eva értéke.

2.6.7. acc

Ha az adott tárgy az ágensek által hordható tárgy, azaz nem kulcs, van valós acc értéke.

2.7. Tárgy nevek

2.7.1. Entity Helmet 1

Ágens elpusztulásakor az adott cellán hagyja hátra, ha van az Inventory-jában. Ládában is található. Ágens által használható felszerelési tárgy, amely pozitív és negatív statisztikát ad, amelyek hozzáadódnak, illetve kivonódnak az ágens jelenlegi statisztikáiból.

- Erő: |2,4|
- Kitérés: |-2,-4|
- Kitartás: |2,4|
- Pontosság: |-2,-4|

2.7.2. Entity Helmet 2

Ágens elpusztulásakor az adott cellán hagyja hátra, ha van az Inventory-jában. Ládában is található. Ágens által használható felszerelési tárgy, amely pozitív és negatív statisztikát ad, amelyek hozzáadódnak, illetve kivonódnak az ágens jelenlegi statisztikáiból.

- Erő: $]-2,-4[$
- Kitérés: $]2,4[$
- Kitartás: $]-2,-4[$
- Pontosság: $]2,4[$

2.7.3. Entity Armor 1

Ágens elpusztulásakor az adott cellán hagyja hátra, ha van az Inventory-jában. Ládában is található. Ágens által használható felszerelési tárgy, amely pozitív és negatív statisztikát ad, amelyek hozzáadódnak, illetve kivonódnak az ágens jelenlegi statisztikáiból.

- Erő: $]2,4[$
- Kitérés: $]-2,-4[$
- Kitartás: $]2,4[$
- Pontosság: $]-2,-4[$

2.7.4. Entity Armor 2

Ágens elpusztulásakor az adott cellán hagyja hátra, ha van az Inventory-jában. Ládában is található. Ágens által használható felszerelési tárgy, amely pozitív és negatív statisztikát ad, amelyek hozzáadódnak, illetve kivonódnak az ágens jelenlegi statisztikáiból.

- Erő: $]-2,-4[$
- Kitérés: $]2,4[$
- Kitartás: $]-2,-4[$
- Pontosság: $]2,4[$

2.7.5. Entity Weapon 1

Ágens elpusztulásakor az adott cellán hagyja hátra, ha van az Inventory-jában. Ládában is található. Ágens által használható felszerelési tárgy, amely pozitív és negatív statisztikát ad, amelyek hozzáadódnak, illetve kivonódnak az ágens jelenlegi statisztikáiból.

- Erő: $]1,5[$

- Kitérés: $]-1,-3[$
- Kitartás: $]0,-1[$
- Pontosság: $]-1,2[$

2.7.6. Entity Weapon 2

Ágens elpusztulásakor az adott cellán hagyja hátra, ha van az Inventory-jában. Ládában is található. Ágens által használható felszerelési tárgy, amely pozitív és negatív statisztikát ad, amelyek hozzáadódnak, illetve kivonódnak az ágens jelenlegi statisztikáiból.

- Erő: $]-1,2[$
- Kitérés: $]1,3[$
- Kitartás: 0
- Pontosság: $]1,5[$

2.7.7. Kulcs

Vagy Ágensektől, vagy celláról szerezhetőek meg. Szerepük az ajtók kinyitása az ágens által. Kulcsot tartalmazó ágens, elpusztulásakor a legutolsó helyén hagyja a kulcsot.

2.8. Ágensek

2.8.1. Statisztika

- Ágensek 4 alap statisztikával rendelkeznek.
- Ez a 4 statisztika a következők: Erő, Kitérés, Kitartás, Pontosság.
- Az ágensek kezdő statisztikája az adott nyílt intervallumon kerül kisorsolásra.

2.8.2. Statisztikák

Kezdő statisztika:

- Erő(str): $]0,1[$
- Kitartás(vit): $]0,1[$
- Kitérés(eva): $]0,1[$
- Pontosság(acc): $]0,1[$

A következő tárgyakkal kezd a szimulációban:

- Entity Helm no.1,
- Entity Armor no.1,
- Entity Weapon no.1.

2.8.3. Statisztikák jelentősége

Statisztikák jelentősége két különböző faction változóval rendelkező ágens összehasonlításánál látszódik meg. Két fontosabb dologban játszik szerepet, az ágens statisztikája a fent említett találkozásnál.

Egy ágens, egy másik ágensre mért csapásának sikeressége függ attól hogy nagyobb-e a támadó ágens acc értéke, mint a támadást elszenvedő fél eva értéke. Ha nagyobb a támadó ágens acc értéke, mint a támadott eva értéke, akkor 100%-osan betalál a sebzése. Ha nem nagyobb, akkor 20% esélye van arra a támadott félnek, hogy ne kapjon semmilyen sebzést.

Egy ágens, egy másik ágensre mért csapásának a sebzése függ attól hogy nagyobb-e a támadó ágens str értéke, mint a támadást elszenvedő fél vit értéke. Ha nagyobb a támadó ágens str értéke, mint a támadott vit értéke, akkor 20% esélye van arra a támadó félnek, hogy dupla sebzést mérjen be. Ha nem nagyobb, akkor fixen 1 sebzést okoz a támadása.

2.8.4. Sebzés számlálás

A támadásnak két végkimenetele lehet:

- Találat,
- Eltévesztve.

Találat esetén, kiszámolódik a sebzés mértéke.

Eltévesztés esetén nem történik meg a sebzés.

2.9. Ágens főbb tulajdonságai

- str(Integer)
Az ágens létrejöttkor kerül definiálásra általunk.
- vit(Integer)
Az ágens létrejöttkor kerül definiálásra általunk.
- eva(Integer)
Az ágens létrejöttkor kerül definiálásra általunk.
- acc(Integer)
Az ágens létrejöttkor kerül definiálásra általunk.
- maxLife(Integer)
Az ágens létrejöttkor kerül definiálásra általunk.
- faction: (Integer)
Lehet 0,1. Fontos változó a combat létrejöttéhez.
- healTurn: (Integer)
Változó, amely azt tárolja, hogy melyik kör végén lesz HP regeneráció.

2.9.1. Kinézetet befolyásoló tényező

- Ágens állása. Látszódik, hogy milyen irányba néz.
- Az ágens faction típusa.

2.9.2. Kommunikáció közöttük

Ágensek bemenete, információ szerzése:

- A hozzájuk szomszédos blockokat vizsgálják, ha nem találnak semmit megvizsgálják az általuk által érzékelhető blockokat.
- A balra, jobbra, felfelé és lefelé irányban érzékelnek 2 blocknyi távolságra.

Ágensek belső memóriája a következőket tartalmazza:

- Inventory állapota.
- Karakter ablak állapota.
- Aktuális HP állapota.
- A blockok hashMap értékei. (hányszor volt az adott blockokon, segít felderíteni a teljes mapot.)
- Észlelt Objektumok helyzete, típusa.
- Inventoryban tárolt itemek.
- Észlelt Ágens kinézete. (ellenséges, vagy nem)
- Észlelt Ágens helyzete. (X,Y koordináta)

2.9.3. Megfigyelési akcióik

Információk feldolgozása:

- Inventory mérete és kihasználtsága, ha elérte a max méretet, nem képes az általa kívánt Tárgyat felvenni.
- Karakter ablakban tárolt statisztikák számossága.
- Aktuális HP mennyisége. 5 Körönként 1 HP visszatöltés.(Lásd: Idő)
- Az adott körben lehetséges lépések hashMap értékei, amelyek az adott block koordinátáit tárolják és azt, hogy az adott ágens hányszor lépett rá az adott blockra. (Szükséges azért, hogy idővel biztosan felfedezze az egész mapot)
- Észlelt Objektumok helyzete, típusa. Ezekből az információkból eldönti a prioritást és meghatározza, hogy mely irányba fog végül haladni.
- A nem használt tárgyak eldobása, ha nem jobb a jelenleg hordottnál, ezáltal biztosítva, hogy ne teljen meg az inventory.
- Ágens grafikai megjelenéséből leszűrt információk számításba vétele. (ellenség-e)
- Támadható Ágensek közül a legkisebb HP-val rendelkező Ágens támadását kezdeményezi.

2.9.4. Ágens tevékenységeinek a köre

Ezek bármelyik használata, felhasználja az ágens körét.

- Lépés.
 - A lépés akkor lehetséges, hogyha az ágens közvetlen közelében, azaz vagy az X vagy az Y koordinátájával szomszédos cellán nem tartózkodik mozgást korlátozó ágens/térbeli objektum. Lásd itt: Mapon található mozgást kizáró objektumok
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.
- Ajtó nyitás.
 - Szükséges az Inventoryban egy Key nevű tárgy, amely a kulcs.
 - Ha az ágens közvetlen közelében létezik a láda, a nyitás akkor lehetséges.
 - Nyitása után eltűnik az ajtó, és a kulcs az Inventoryból.
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.
- Láda nyitás.
 - Ha az ágens celláján létezik a láda, a nyitás akkor lehetséges.
 - Kinyitása, ha nincs támadható ágens, után győzelem.
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.
- Felvétel.
 - Ha az Inventoryban van hely, akkor lehetséges.
 - Ha az ágens celláján létezik valamilyen tárgy, ekkor a tárgyat a felvétel akcióval eltünti a celláról és az inventoryba kerül.
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.
- Törlés.
 - Ha az Inventoryban van olyan tárgy, amely nem kulcs és nem hordott az ágens által, akkor lehetséges.
 - Az ágens Inventoryjából kitörli az adott nem használt tárgyat.
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.
- Felszerelés.
 - A felszerelés mindig lehetséges az ágens saját körében.

- Megvizsgálja, hogy az utoljára felvett tárgy statisztikái jobbak-e, mint az ő általa hordott azonos típusú hordható tárgy.
 - Ha igen, felveszi azt, ezáltal megválasztotva saját statisztikáit.
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.
- Támadás.
 - A támadás akkor lehetséges, hogyha az ágens közvetlen közelében, azaz vagy az X vagy az Y koordinátájával szomszédos cellán tartózkodik egy ellenséges ágens.
 - Több ellenséges ágens esetén, a legkisebb HP-val rendelkező fogja támadni.
 - Ez az akció az ágens körének felhasználást igényli, ahhoz hogy végrehajtsa azt.

2.9.5. Ágens fő céljai

- Megtalálni az ellenséges Chestet.
- Elpusztítani az ellenséges ágenseket.

2.9.6. Ágens másodlagos céljai

- Tárgy felvétel,
- Tárgy hordása adott esetekben,
- Ajtó kinyitás,
- Inventory menedzsment,
- Mindig a kevesebbszer bejárt blockokat választani mozgásakor.

2.9.7. Ágens függvény

A következő ábrán látható az ágens cselekvésének meghatározására szolgáló ciklus, amely a szimuláció kezdetétől működik. ??

A csomópontok kifejtése:

- wantToAttack?
 - Ha közvetlen közelében van egy ellenséges Ágens, mindig azt az akciót választja, ahol támadást mér be az ellenséges Ágensnek.
- wantToEquip?
 - Ha a legutoljára általa felvett tárgy statisztikái több statot adnak összességében, mint az általa hordott azonos típusú tárgy, akkor lecseréli.
- wantToPickUp?

- Ha az ágenset tartalmazó block-on létezik valamilyen felvehető tárgy az ágens által, akkor felveszi.
- wantToDeleteAnItem?
 - Ha az ágens inventoryja tartalmaz olyan hordható tárgyat, amely nem jobb mint az általa hordott tárgy, akkor eltávolítja az inventoryjából.
- wantToOpenDoor?
 - Ha van kulcs nála és a közvetlen közelében van egy ajtó, akkor az ajtó kinyitását választja.
- wantToMoveToItem?
 - Ha van olyan közvetlen közeli block az ágens szomszédjában, amelyre lehetséges a lépés és tartalmaz valamilyen felvehető tárgyat, akkor összeveti az Ágens és a felvenni kívánt tárgy koordinátáit, és az alapján eldönti milyen irányba mozogjon az ágens.
- newEnemy?
 - Ez az eset akkor lép fel, ha semmilyen más cselekvést nem akart elvégezni az adott körben az Ágens és lát a detectableBlocks-on belül olyan blockot, amely egy ellenséges Ágenst tartalmaz, ilyenkor a hozzá közeli blockot választja ki következő lépésének.
- newItem?
 - Ez az eset akkor lép fel, ha semmilyen más cselekvést nem akart elvégezni az adott körben az Ágens és lát a detectableBlocks-on belül olyan blockot, amely egy kívánt tárgyat tartalmaz, ilyenkor a hozzá közeli blockot választja ki következő lépésének.
- wantToMove?

Ez az eset akkor lép fel, ha semmilyen más cselekvést nem akart elvégezni az adott körben az Ágens.

 - A lehetséges blockok közül kiválasztja azt a blockot a következő lépésének, amelyen még nem volt soha.
 - Ha több olyan lehetséges block van, amelyre léphet, akkor véletlenszerűen választ egyet a lehetőségek közül.
 - Ha nincs egy olyan block se, ahol ne lett volna már, akkor a legkevesebbszer bejárt blockot fogja választani.
 - Ha több legkevesebbszer bejárt block létezik a lehetséges blockok közül, akkor véletlenszerűen választ egyet a lehetőségek közül.

2.10. Cél Tárgy

Map teljesítéséhez szükséges tárgy, amely egy Chest. A Kék faction-nek a kék chestet, a piros faction-nek a piros chestet kell elérnie. A mapon egy szobában helyezkedik el.

2.11. Combat

A szimulációban 1-es faction változójú Ágens és 2-es faction változójú Ágens közötti HP elvétel lehetséges,

- szomszédos cellából támadás által,
- a támadás lehet betalált, elvétett,
- a támadás nagysága lehet normális vagy nagy.

2.12. Interface felület

2.12.1. Pause

- prekondíció: A szimulációs ablakban vagyunk.
- általános működés: Megnyomjuk a P gombot.
- alternatív esetek: Rossz tevékenység történik be.
- Postkondíció: Megjelent a Paused kiírás bal fent.
- kivételes esetek: Nem ált meg a játékmenet.

2.12.2. Inventory

- prekondíció: A szimulációs ablakban vagyunk és meg van már nyitva a karakter ablak.
- általános működés: Megnyomjuk az I gombot.
- alternatív esetek: Rossz ablak nyílik meg.
- Postkondíció: Megnyílt az inventory ablak.
- kivételes esetek: Nem nyílt meg az ablak.

2.12.3. Karakter

- prekondíció: A szimulációs ablakban vagyunk.
- általános működés: Megnyomjuk a C gombot.
- alternatív esetek: Rossz ablak nyílik meg.
- Postkondíció: Megnyílt a Karakter ablak.
- kivételes esetek: Nem nyílt meg az ablak.

2.13. Mapon található mozgást kizáró objektumok

2.13.1. Brick

Az Brick egy oszlop, amely a map generáláskor kerülnek bizonyos cellákra. Oszlopoknak semmilyen különleges funkciója nincs azonkívül, hogy map komplexitását kívánja növelni azzal, hogy mozgást és látást korlátozó szerepet lát el.

2.13.2. Zárt ajtó

Az ajtók map generáláskor kerülnek csak olyan cellákra, ahol egy szobából egy folyosón lehetséges elhagyni a szobát. Mindig zárt alakban kerül az adott cellákra, kinyitásához nem szükséges semmilyen különleges tárgy. Ha bármely karakter egy ajtót tartalmazó cellára lép, az ajtó kinyílik amíg Entityt tartalmaz az adott cella. Ahogyan az Oszlopok, a zárt ajtók is map komplexitását kívánja növelni azzal, hogy látást korlátozó szerepet lát el.

2.14. Idő

Az idő a szimuláción belül minden ágens által végrehajtott akció által telik. Minden 5. kör után, 1 HP-t töltenek vissza az ágensek, ha kisebb az aktuális életerejük, mint a maximum. Ha egy Ágens elvégez egy akciót, akkor az irányítás átkerül egy másik Ágensre.

2.15. Kamera

A kamera block-ról blockra képes haladni 3 koordináta meglépésével. A kamera képes mozogni, akkor is ha a játékmenet szünetel.(Paused)

2.16. Irányítás

Szimulációban használható irányítás:

- Kamera mozgatása: WASD
- Ágens statisztika ablak megnyitása/bezárása: C
- Ágensek statisztika ablaka közötti lépegetés: Q(hátra), E(előre)
- Adott Ágens statisztika oldalhoz Inventory megnyitása/bezárása: I



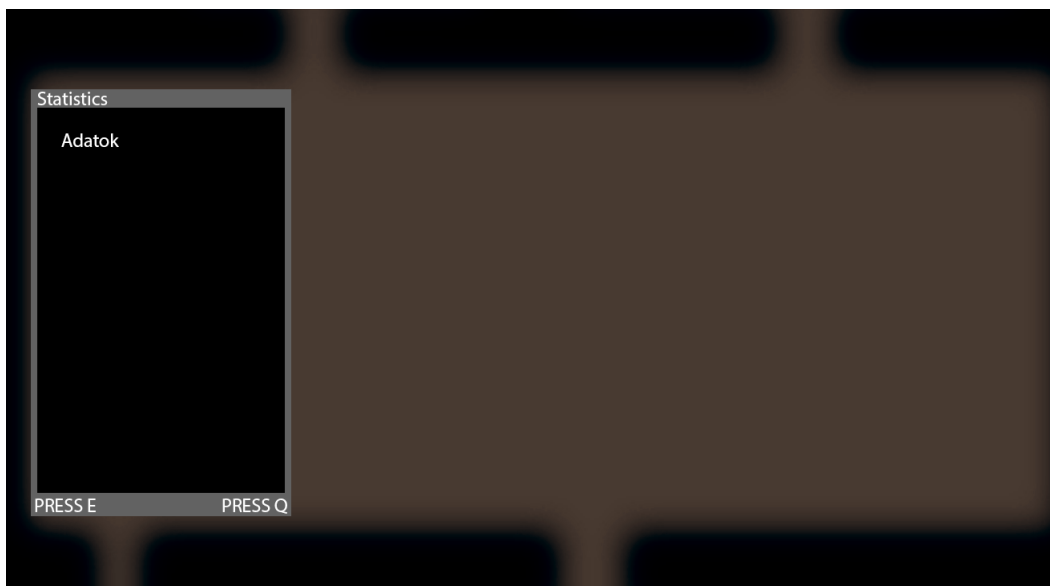
2.1. ábra. nonPaused



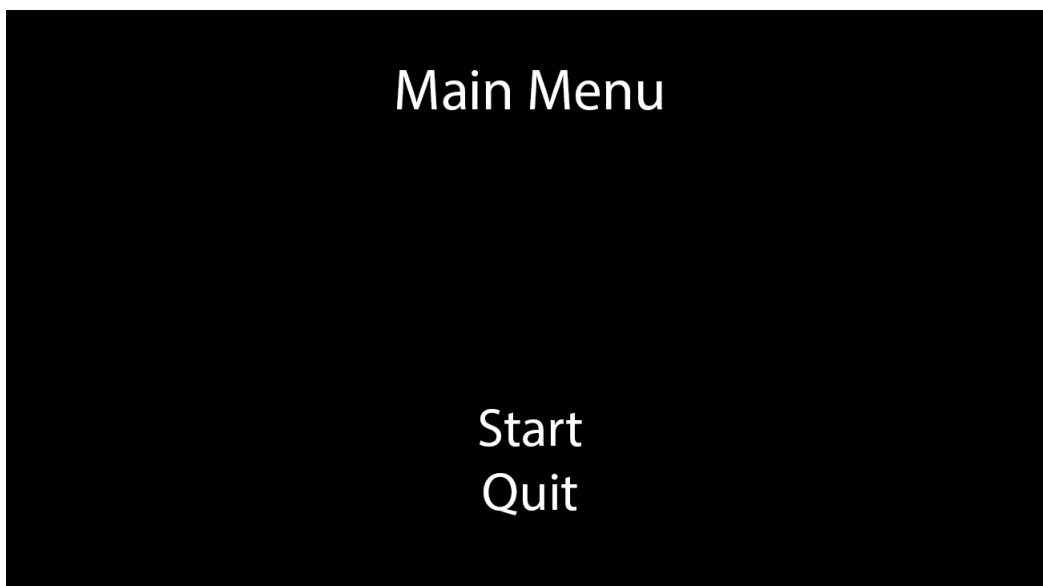
2.2. ábra. Paused



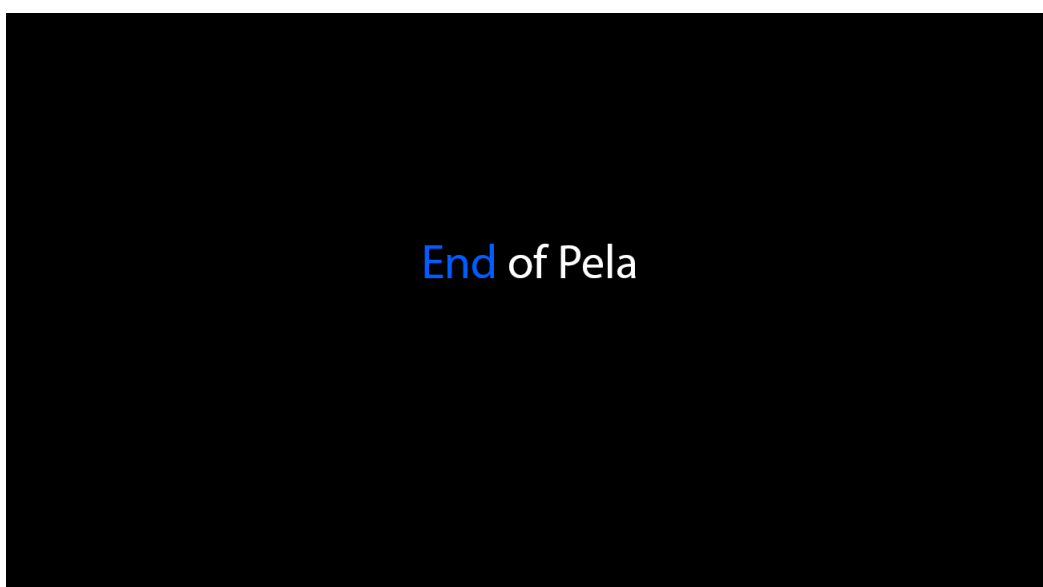
2.3. ábra. Inventory



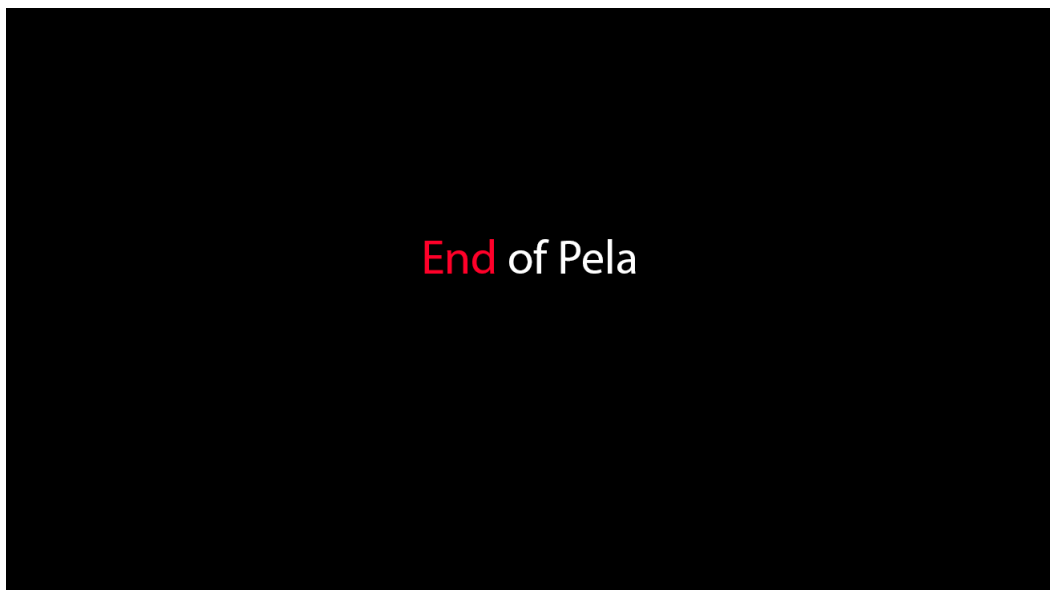
2.4. ábra. Statistics



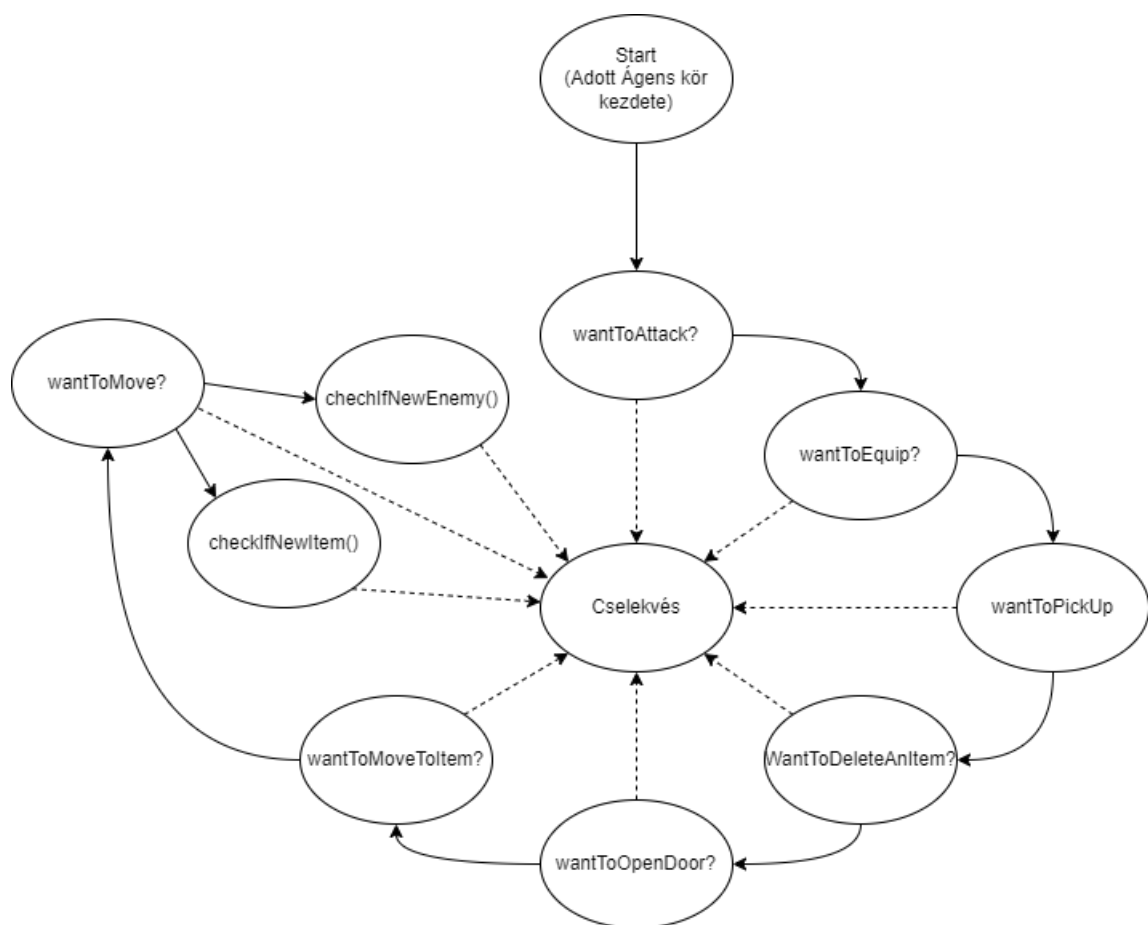
2.5. ábra. Main Menu



2.6. ábra. Blue End Screen



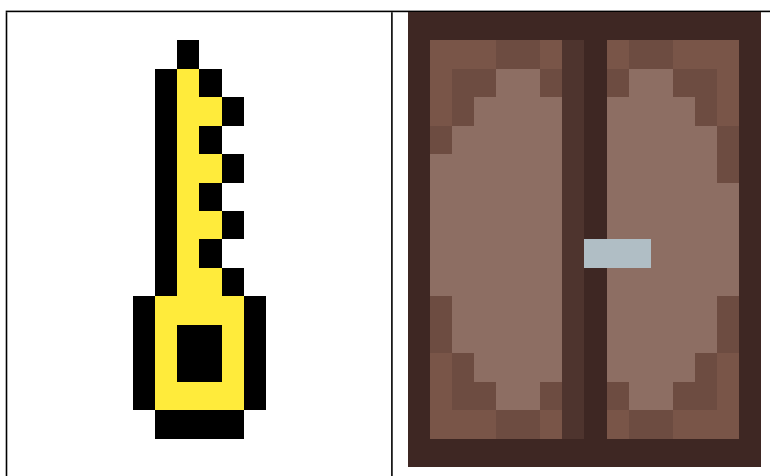
2.7. ábra. Red End Screen



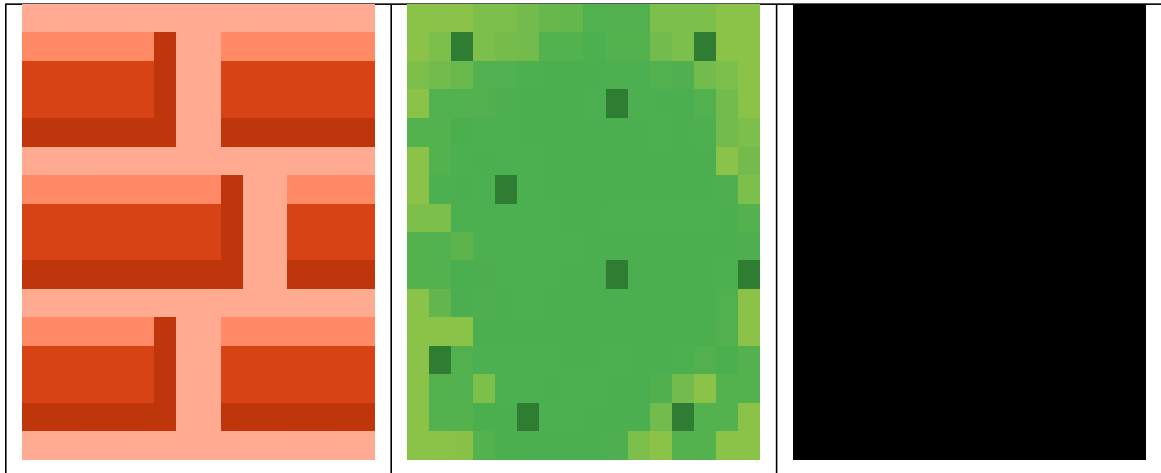
2.8. ábra. Agent Action Priority



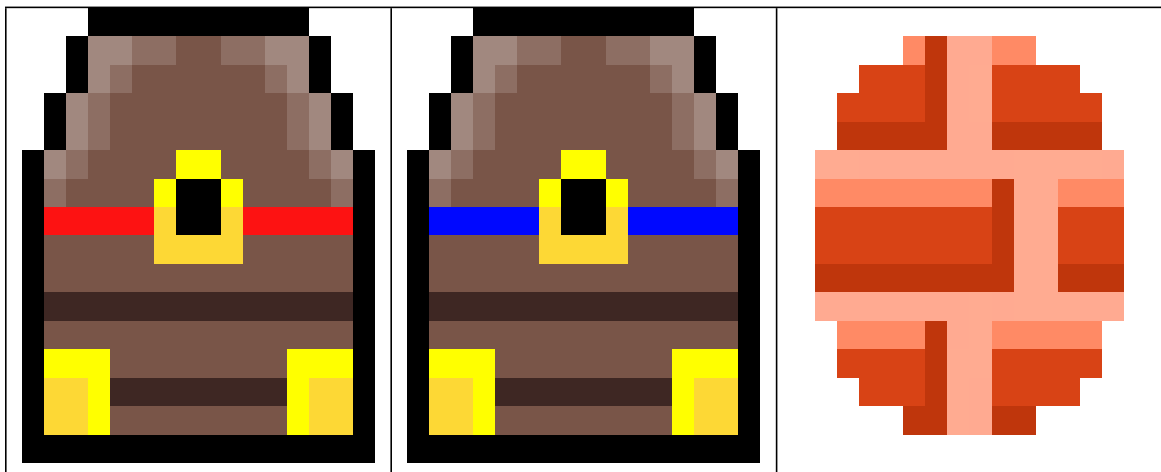
2.1. táblázat. Item set 01



2.2. táblázat. Key and Door



2.3. táblázat. Blockok



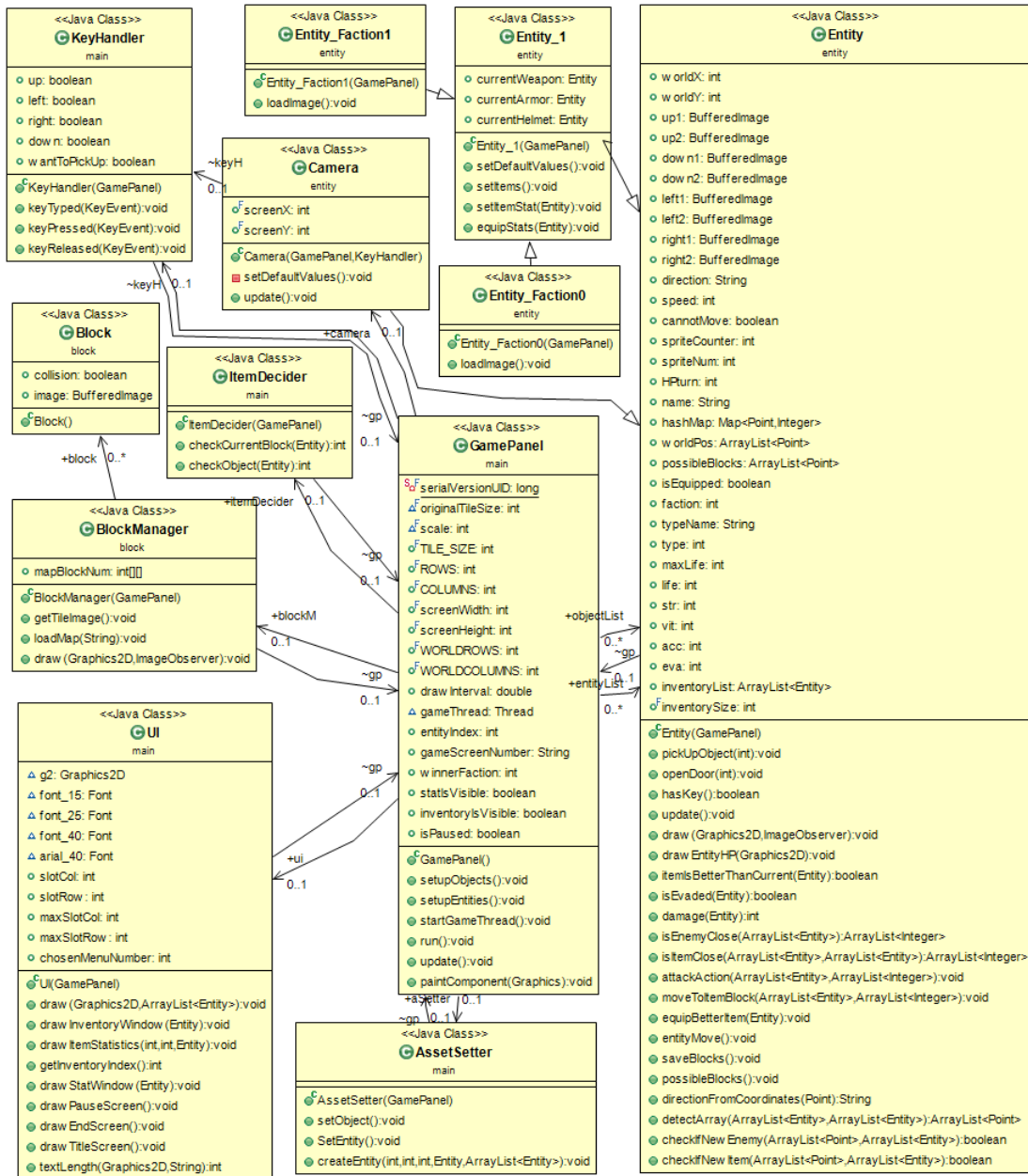
2.4. táblázat. Chests and Brick

3. fejezet

Tervezés

A szimulációs környezet megfelelő működéséhez el kell tárolnunk az ágensek és a mapot összeállító blockok adatait. Ennek a megvalósításához szükséges leírni mely adatokat kell eltárolnunk és adatmodelleket készítése is elegendhetetlen az átláthatóságért.

3.1. Adatmodellek



3.1. ábra. UML

4. fejezet

Megvalósítás

Ebben a fejezetben a ágens heurisztikus mozgását és akcióit végrehajtó biztosító függvények bemutatása történik meg röviden.

```
if(wantToAttack) {
    attackAction(gp.entityList, isEnemyCloseIndexList);
} else if(wantToEquip) {
    equipBetterItem(inventoryList.get(inventoryList.size()
    -1));
} else if(wantToPickUp) {
    pickUpObject(curObjIndex);
} else if(wantToDeleteAnItem) {
    System.out.println("Deleted item: "
        + inventoryList.get(deleteIndex).name);
    inventoryList.remove(deleteIndex);
} else if(wantToOpenDoor) {
    openDoor(doorIndex);
} else if(wantToMoveToItem) {
    if(cannotMove==false) {
        moveToItemBlock(gp.objectList,
            isItemCloseIndexList);
        entityMove();
    } else System.out.println("Skipped Turn");
} else if(wantToMove) {
    if(cannotMove==false) {

        ArrayList<Point> detectableBlocks =
            detectArray(gp.entityList, gp.objectList);

        if(!checkIfNewEnemy(detectableBlocks,
            gp.entityList)) {
            if(!checkIfNewItem(detectableBlocks,
                gp.objectList)) {
                possibleBlocks();
            }
        }
    }
}
```



```

    }

    entityMove ();
} else System.out.println("Skipped Turn");

}

```

4.1. wantToAttack

Boolean típusú változó, amely akkor igaz, ha közvetlen szomszédos blockokon létező ellenséges ágensok száma nagyobb, mint 0.

Igaz érték esetén meghívódik az `attackAction()` függvény, amelynek paramétere az ágensokból álló lista és az `isEnemyCloseIndexList`, amely a közeli blockon lévő ellenséges ágensok közül a legkevesebb HP-val rendelkező ágens indexét adja vissza az ágens listából.

Az `attackAction()` függvény beállítja az adott ágens irányát, arra amelyik irányba hajta végre a támadását és elvégzi a támadást.

Támadás előtt először megvizsgálja betalált-e a találat, ha betalált, akkor meghívódik a `damaga()` függvény, amely kivonja az adott sebzést az adott ágenstől, amelyik elszenvedte a támadást.

4.2. wantToEquip

Boolean típusú változó, amely akkor igaz, ha az adott ágens inventoryjában az utoljára felvett tárgy hordható típusú és összességében több statisztikát ad, mint az ágens által jelenleg hordott azonos típusú tárgy.

Az `equipBetterItem()` függvénynek egy paramétere van, amely egy Integer szám, ami az adott ágens inventoryjának utolsó használt slotja, amely a legutoljára felvett tárgy. Meghívásakor leszereli az eddig hordott tárgyat, és felszereli az újonnan felvett tárgyat. Statisztikák ablakon nyomon lehet követni ezt a változást.

4.3. wantToPickUp

Boolean típusú változó, amely akkor igaz, ha létezik valamilyen tárgy az ágens jelenlegi blockján.

Ha igaz értéket ad vissza, akkor meghívódik a `pickUpObject()` függvény, amelynek paramétere az adott blockon lévő tárgy indexe a tárgyak listájában. Meghívásakor az adott tárgy bekerül az ágens első nem használt inventory slotjába.

4.4. wantToDeleteAnItem

Boolean típusú változó, amely akkor igaz, ha létezik valamilyen nem viselt tárgy az ágens inventoryjában.

Igaz érték esetén megtörténik a `deletedIndex` által tárolt elsőnek talált nem használt tárgy indexének a törlése az inventoryjából, amely nem az ágens által hordott és nem kulcs.

4.5. `wantToOpenDoor`

Boolean típusú változó, amely akkor igaz, ha létezik ajtó az ágens szomszédos block-jában és van az ágens inventoryjában legalább 1 kulcs.

Az `openDoor()` függvénynek egy paramétere van, amely az ajtó indexe a tárgy listában. Meghívásakor az ajtó objektum eltűnik, és eltávolítja a felhasznált kulcsot az ágens inventoryjából.

4.6. `wantToMoveToItem`

Boolean típusú változó, amely akkor igaz, ha a közeli blockokon lévő tárgyak listája nagyobb, mint 0 és az ágens inventoryja nincs tele.

Valódi ágens mozgás csak akkor jön létre, ha van olyan szomszédos block, amelyre képes lépni, ezt a `CannotMove` false értéke biztosítja.

Majd meghívódik a `moveToItemBlock()`, amelynek két paramétere van, az objektum lista és a szomszédos közelében lévő tárgyak indexének a listája. Ez a függvény beállítja az ágens `direction` értéket.

Majd meghívódik az `entityMove()` függvény, amely az ágens által választott irányba lép előre egyet és vagy felrakja a `hashMap`-re, vagy növeli az értékét.

4.7. `wantToMove`

Boolean típusú változó, amely mindig igaz, ha minden fentebb sorolt boolean változó hamis, ebben a pontban összefoglalt akció fog megtörténni.

Valódi ágens mozgás csak akkor jön létre, ha van olyan szomszédos block, amelyre képes lépni, ezt a `CannotMove` false értéke biztosítja.

A pontokat tároló listában a `detectArray()` függvény minden olyan `x,y` párost átad, amely az ágenstől vízszintes vagy függőlegesen 2 blocknyira van és tartalmaz valamilyen kívánt tárgyat. A `detectArray` két paramétere az ágens listája és az objektum lista.

A `checkIfNewEnemy()` függvény paramétere a pontokat tároló lista és az ágens listája. Ha talál valamilyen ellenséges ágenst a vizsgálandó pontokon, akkor azt az irányt fogja beállítani az ágensnek, amely az ellenséges ágens felé néz.

Utána meghívódik az `entityMove()` függvény, amely a kiválasztott irányba lép egyet és vagy felrakja a `hashMap`-re, vagy növeli az értékét.

Ha a `checkIfNewEnemy()` hamis értéket ad vissza, akkor megvizsgálja a `checkIfNewItem()` függvényt, amelynek két paramétere van, a vizsgálandó pontokat tároló lista és az objektumok listája.

Ha talál valamilyen számára érdekes tárgyat a vizsgálandó pontokon, akkor azt az irányt fogja beállítani az ágensnek, amely a kívánt tárgy felé néz.

Ha a `checkIfNewItem()` függvény is hamis értéket ad vissza, akkor meghívódik a `possibleBlocks()` függvény, amely felel a map felfedezéséért.

A függvény megvizsgálja a lehetséges lépéseket, hogy van-e köztük olyan block, amelyen még egyszer sem járt az ágens. Ha több ilyen van, akkor véletlenszerűen választ egyet azok közül.

Ha már minden lehetséges blockon járt legalább egyszer, akkor a lehetséges lépések közül a legkevesebbszer bejárt blockot választja. Ha több ilyen block is van, amin ugyanannyiszor volt, akkor véletlenszerűen választ közülük egyet.

5. fejezet

Tesztelés

A fejezetben be kell mutatni, hogy az elkészült alkalmazás hogyan használható. (Az, hogy hogyan kell, hogy működjön, és hogy hogy lett elkészítve, az előző fejezetekben már megtörtént.)

Jellemzően az alábbi dolgok kerülhetnek ide.

- Tesztfuttatások. Le lehet írni a futási időket, memória és tárigényt.
- Felhasználói kézikönyv jellegű leírás. Kifejezetten a végfelhasználó szempontjából lehet azt bemutatni, hogy mit hogy lehet majd használni.
- Kutatás kapcsán ide főként táblázatok, görbék és egyéb részletes összesítések kerülhetnek.

6. fejezet

Összefoglalás

Hasonló szerepe van, mint a bevezetésnek. Itt már múltidőben lehet beszélni. A szerző saját meglátása szerint kell összegezni és értékelni a dolgozat fontosabb eredményeit. Meg lehet benne említeni, hogy mi az ami jobban, mi az ami kevésbé jobban sikerült a tervezettnél. El lehet benne mondani, hogy milyen további tervek, fejlesztési lehetőségek vannak még a témával kapcsolatban.

Irodalomjegyzék

CD Használati útmutató

Ennek a címe lehet például *A mellékelt CD tartalma* vagy *Adathordozó használati útmutató* is.

Ez jellemzően csak egy fél-egy oldalas leírás. Arra szolgál, hogy ha valaki kézhez kapja a szakdolgozathoz tartozó CD-t, akkor tudja, hogy mi hol van rajta. Jellemzően elég csak felsorolni, hogy milyen jegyzékek vannak, és azokban mi található. Az elkészített programok telepítéséhez, futtatásához tartozó instrukciók kerülhetnek ide.

A CD lemezre mindenképpen rá kell tenni

- a dolgozatot egy `dolgozat.pdf` fájl formájában,
- a LaTeX forráskódját a dolgozatnak,
- az elkészített programot, fontosabb futási eredményeket (például ha kép a kimenet),
- egy útmutatót a CD használatához (ami lehet ez a fejezet külön PDF-be vagy Markdown fájlként kimentve).