**RÉSZLETES TERVEK**

40 – ZETA

Konzulens:

KOVÁCS BOLDIZSÁR

Csapattagok

|  |  |  |
| --- | --- | --- |
| Alpek Dávid Zsolt | C31X0F | alpek.david.zsolt@gmail.com |
| Csia Klaudia Kitti | HA5YCV | kitkat@sch.bme.hu |
| Litavecz Marcell | IPHJNB | marcell.litavecz@gmail.com |
| Marton Judit | M0MYIM | judit.marton@edu.bme.hu |
| Ruskó Eszter | H8IBRC | eszter@rusko.hu |

2022. 04. 12.

# Részletes tervek

## Osztályok és metódusok tervei

### Equipment

#### Felelősség

Az óvóhelyen megszerezhető eszközöket reprezentáló absztrakt osztály.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **String ID**: Az Equipment objektum egyedi azonosítóját tárolja. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: absztrakt metódus. **Láthatóság:** +
* **String getID()**: Visszaadja az ID attribútum értékét. **Láthatóság:** +

### Gloves

#### Felelősség

Speciális típusú felszerelés, amellyel a virológus visszadobhatja a rákent ágenst a támadóra. Viszont ezt csak meghatározott alkalommal használhatja. Továbbá a kesztyű óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equipment

#### Interfészek

Nincs

#### Attribútumok

* **int Usable**: Tárolja, hogy kesztyű hány alkalommal használható. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a kesztyűt a kapott virológus inventoryjának kesztyű listájához. **Láthatóság:** +
* **void use(Virologist vFr, Virologist vTo, Agent a)**: Meghívja a megtámadott virológus támadás függvényét a támadó virológussal és a használt ágenssel. **Láthatóság:** +
* **boolean wannaUse()**: Megkérdezi a játékost, hogy akarja-e használni a kesztyűt. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** +
* **void setUsable(int NumOfTimes)**: Beállítja a Usable attribútum értékét a paraméterként megkapott értékre. **Láthatóság:** +
* **int setUsable(int NumOfTimes)**: Visszaadja a Usable attribútum értékét. **Láthatóság:** +

### Sack

#### Felelősség

Speciális típusú felszerelés, amellyel a virológus inventoryjának maximális anyagtárolóképessége megnő. Továbbá a zsák óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equpment

#### Interfészek

Nincs

#### Attribútumok

* **int Capacity**: Tárolja, hogy a zsák mennyivel növeli meg a maximális anyagtárolóképességet, csak olvasható. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a zsákot a kapott virológus inventoryjának zsák listájához. **Láthatóság:** +
* **int getCapacity()**: Visszaadja a Capacity attribútum értékét. **Láthatóság:** +

### Cloak

#### Felelősség

Speciális típusú felszerelés, amely egy adott százalékos aránnyal megakadályozza, hogy a köpenyt viselő virológusra hatással legyen a felkent ágens. Továbbá a köpeny óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equipment

#### Interfészek

Nincs

#### Attribútumok

* **double EvadePercentage**: Tárolja, hogy a köpeny milyen százalékkal védi ki a felkent ágens hatását. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a köpenyt a kapott virológus inventoryjának köpeny listájához. **Láthatóság:** +
* **boolean evade()**: EvadePercentage százalékkal igazat, egyébként hamisat ad vissza. **Láthatóság:** +

### Axe

#### Felelősség

Speciális típusú felszerelés, amellyel a virológus meg tud ölni egy másik virológust. Továbbá a fejsze óvóhelyről inventoryba való felvételéért felelős.

#### Ősosztályok

Equipment

#### Interfészek

Nincs

#### Attribútumok

* **boolean Used:** Tárolja, hogy a fejszét használták-e már. **Láthatóság:** -

#### Metódusok

* **void pickUp(Virologist v)**: Hozzáadja a fejszét a kapott virológus inventoryjának fejsze listájához. **Láthatóság:** +
* **void use(Virologist vTo)**: Meghívja a kapott virológus die metódusát és igazra állítja a Used attribútumot. **Láthatóság:** +
* **boolean wannaUse():** **()**: Megkérdezi a játékost, hogy akarja-e használni a fejszét. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** +
* **boolean getUsed():** Visszaadja a Used attribútum értékét. **Láthatóság:** +

### Inventory

#### Felelősség

A virológus által óvóhelyről, raktárból, laborból megszerzett eszközöket, anyagokat, genetikai kódokat tárolja, minden típust egy külön listában. Azok az eszközöket, amelyeket a virológus levett magáról szintén a megfelelő eszköz listába kerül vissza. Illetve nyilvántartja, hogy maximum hány anyag és mennyi eszköz lehet az inventoryban.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **int maxMaterial**: Tárolja, hogy maximum mennyi eleme lehet az anyaglistáknak összesen. **Láthatóság:** -
* **int maxEquipment**: Tárolja, hogy maximum mennyi eleme lehet az eszközlistáknak összesen, csak olvasható. **Láthatóság:** -
* **ArrayList** **<Gloves> storedGloves**: Tárolja a virológus által óvóhelyről felvett kesztyűket, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<Sack> storedSacks**: Tárolja a virológus által óvóhelyről felvett zsákokat, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<Cloak> storedCloaks**: Tárolja a virológus által óvóhelyről felvett köpenyeket, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<Axe> storedAxes**: Tárolja a virológus által óvóhelyről felvett fejszéket, és azokat, amelyeket a virológus levett magáról. **Láthatóság:** -
* **ArrayList** **<GCode> learntGCode**: Tárolja a virológus által laborból megtanult genetikai kódokat, minden tárolt kód különböző. **Láthatóság:** -
* **ArrayList** **<Aminoacid> aminos**: Tárolja a virológus által raktárból felvett aminosavakat. **Láthatóság:** -
* **ArrayList** **<Nukleotid> nukleotids**: Tárolja a virológus által raktárból felvett nukleotidokat. **Láthatóság:** -

#### Metódusok

* **void init()**: Inicializálja a 7 darab tárolt listát, és beállítja a maxMaterial-t 20-ra. **Láthatóság:** -
* **void setMaxMaterial(int maxMaterial)**:  
  Ha a jelenlegi maxMaterial nagyobb mint a kapott maxMaterial, akkor először ki kell számolni a túlcsordulást. Azaz, hogy a tárolt anyaglisták hosszainak összegee és a kapott maxMatrial különbsége (túlcsordulás = aminos lista hossza+ nukleotids lista hossza – maxMaterial). Ha a túlcsordulás nagyobb, mint nulla, akkor ellenőrizni kell, hogy az anyaglisták közül melyik a hosszabb és abból egy ciklus segítségével egyesével törlni kell annyi elemet amennyi a túlcsordulás volt.  
  Egyébként a maxMaterial értékét beállítja a kapott értékre.  
  **Láthatóság:** +
* **int getMaxMaterial()**: Visszaadja a maxMaterial értékét. **Láthatóság:** +
* **void addGloves(Gloves g)**: Ha a kapott Gloves objektum nem null és felvehető eszköz az inventoryba, akkor a tárolt kesztyűk listához hozzáadja a g-t. **Láthatóság:** +
* **void addSack(Sack s)**: Ha a kapott Sack objektum nem null és felvehető az eszköz az inventoryba, akkor a tárolt zsákok listához hozzáadja az s-t. **Láthatóság:** +
* **void addCloak(Cloak c)**: Ha a kapott Cloak objektum nem null és felvehető eszköz az inventoryba, akkor a tárolt köpenyek listához hozzáadja a c-t.  
  **Láthatóság:** +
* **void addAxe(Gloves a)**: Ha a kapott Axe objektum nem null és felvehető eszköz az inventoryba, akkor a tárolt fejszék listához hozzáadja az a-t. **Láthatóság:** +
* **Gloves removeGloves()**: Ha tárolt kesztyűk lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **Sack removeSack()**: Ha a tárolt zsákok lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **Cloak removeCloak()**: Ha a tárolt köpenyek lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **Axe removeAxe()**: Ha a tárolt fejszék lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként nulllal tér vissza. **Láthatóság:** +
* **int learnGCode(GCode g)**: Ha a kapott GCode objektum nem null és a genetikai kód még nincs benne a megtanult genetikai kódok listájában, akkor hozzáadja ehhez a listához a g-t. Majd visszatér a megtanult genetikai kódok listájának hosszával. **Láthatóság:** +
* **void forgetGCode()**: Kiüríti a megtanult genetikai kódok listáját. **Láthatóság:** +
* **void addAmino(Aminoacid am)**: Ha a kapott Aminoacid objektum nem null és felvehető anyag az inventoryba, akkor hozzáadja am-t az aminosavak listájához. **Láthatóság:** +
* **void addNukleo(Nukleotid n)**: Ha a kapott Nukleotid objektum nem null és felvehető anyag az inventoryba, akkor hozzáadja n-t a nukleotidok listájához. **Láthatóság:** +
* **Aminoacid removeAmino()**: Ha az aminosavak lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként null-al tér vissza. **Láthatóság:** +
* **Nukleotid removeNukleo()**: Ha a nukleotidok lista hossza nagyobb, mint nulla, akkor töröl egy elemet a listából és visszatér ezzel az elemmel. Egyébként null-al tér vissza. **Láthatóság:** +
* **int getStoredAminoCount()**: Visszaadja az aminosavak lista hosszát. **Láthatóság:** +
* **int getStoredNukleotidCount()**: Visszaadja a nukleotidok lista hosszát.  
  **Láthatóság:** +
* **boolean canAddEq()**: Megadja, hogy felvehető-e eszköz az inventoryba. Ha a tárolt eszközlisták hosszainak összege kisebb, mint a maxEquipment (tárolt kesztyűk lista hossza + tárolt zsákok lista hossza + tárolt köpenyek lista hossza + tárolt fejszék lista hossza < maxEquipment), akkor igazzal tér vissza. Egyébként hamissal. **Láthatóság:** -
* **boolean canAddMaterial()**: Megadja, hogy felvehető-e anyag az inventoryba. Ha a tárolt anyaglisták hosszainak összege kisebb, mint a maxMaterial (aminosavak lista hossza + nukleotidok lista hossza < maxMaterial), akkor igazzal tér vissza. Egyébként hamissal. **Láthatóság:** -

### Material

#### Felelősség

A raktárból megszerezhető anyagokat reprezentáló absztrakt osztály.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void pickUp(Virologist v)**: absztrakt metódus. **Láthatóság:** +

### Aminoacid

#### Felelősség

Speciális típusú anyag, amely szükséges ágens készítéshez. Az aminosav raktárból inventoryba való felvételét végzi.

#### Ősosztályok

Material

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void pickUp(Virologist v)**: Ha a kapott virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor elkéri a virológustól az inventoryját és meghívja az inventory addAmino metódusát önmagát átadva. **Láthatóság:** +

### Nukleotid

#### Felelősség

Speciális típusú anyag, amely szükséges ágens készítéshez. A nukleotid raktárból inventoryba való felvételét végzi.

#### Ősosztályok

Material

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void pickUp(Virologist v)**: Ha a kapott virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor elkéri a virológustól az inventoryját és meghívja az inventory addNukleo metódusát önmagát átadva. **Láthatóság:** +

### GCode

#### Felelősség

A genetikai kódot reprezentáló osztály. Tárolja, hogy milyen ágens, milyen anyagok felhasználásával készíthető el belőle. Elkészíti az ágenst a virológusnak.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **String GCName**: Tárolja, hogy milyen ágens készíthető el ebből az objektumból. **Láthatóság:** -
* **int NeededAmino**: Tárolja, hogy mennyi aminosav szükséges az ágens elkészítéséhez. **Láthatóság:** -
* **int NeededNukleo**: Tárolja, hogy mennyi nukleotid szükséges az ágens elkészítéséhez. **Láthatóság:** -

#### Metódusok

* **String getGCName()**: Visszaadja, hogy milyen ágens készíthető ebből a genetikai kódból. **Láthatóság:** +
* **Agent create(Virologist v)**: Ha a virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor először elkéri a virológustól az inventoryját. Majd, ha az inventoryban tárolt anyagok mennyisége nagyobb vagy egyenlő, mint a szükséges anyagok (tárolt aminosav mennyiség >= szükséges aminosav és tárolt nukleotid mennyiség >= szükséges nukleotid), akkor egy-egy ciklus segítségével kiveszi az inventoryból a szükséges mennyiséget mind a két anyagból. Ezután egy switch case szerkezettel a GCName alapján elkészíti a megfelelő ágens objektumot és ezzel visszatér.

Egyébként nulllal tér vissza.  
**Láthatóság:** +

### Field

#### Felelősség

Ez az osztály az összes mező ősosztály és sima mezőként működik. Tárolja a szomszédos mezőket és az itt tartózkodó virológusokat. Kezeli a virológus megérkezését a mezőre.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

* **ArrayList** **<Field> Neighbours**: Tárolja a szomszédos mezőket. **Láthatóság:** -
* **ArrayList** **<Virologist> Viros**: Tárolja a mezőn tartózkodó virológusokat. **Láthatóság:** -
* **String ID**: A field objektum egyedi azonosítóját tárolja. **Láthatóság:** -

#### Metódusok

* **void setNeighbours(Field f1, Field f2)**: Ha egyik Field objektum se null, akkor az f1 szomszéd listájához hozzáadja f2-t és az f2 szomszéd listájához hozzáadja f1-t. **Láthatóság:** +
* **ArrayList** **<Virologist> getVirologists()**: Visszaadja a mezőn tartózkodó virológusok listáját. **Láthatóság:** +
* **ArrayList** **<Field> getNeighbours()**: Visszaadja a mező szomszédainak listáját. **Láthatóság:** +
* **void arrive(Virologist v)**: Ha kapott virológus nincs lebénulva, nem vitustáncol és nem medvetáncol, akkor egy ciklussal megnézi, hogy a többi itt található virológus közül le van-e bénulva valamelyik. Ha talál egyet akkor megkérdezi a felhasználót, hogy meg akarja-e lopni (wannaSteal). Ha a válasz igen akkor meghívja a bénult virológus stealBy függvényét és átadja magát, majd kilép ciklusból. **Láthatóság:** +
* **boolean wannaSteal()**: Megkérdezi a játékost, hogy akar-e lopni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **void addVirologist(Virologist v)**: Hozzáadja a kapott Virologist objektumot az itt tartózkodó virológusok listájához. **Láthatóság:** +
* **void removeVirologist(Virologist v)**: Eltávolítja a kapott Virologist objektumot az itt tartózkodó virológusok listájából. **Láthatóság:** +
* **ArrayList** **<Virologist> getVirologists()**: Visszaadja a Viros listát. **Láthatóság:** +
* **String getID()**: Visszaadja az ID attribútum értékét. **Láthatóság:** +

### Storage

#### Felelősség

Speciális típusú mező, amelyen anyagokat szedhet fel az ide érkező virológus.

#### Ősosztályok

Field

#### Interfészek

Steppable

#### Attribútumok

* **boolean Empty**: Tárolja, hogy a raktár éppen üres-e vagy nem. Hamisra van inicializálva. **Láthatóság:** -
* **int remnaingTime**: Tárolja, hogy a raktár, hány körön keresztül lesz üres. **Láthatóság:** -

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát és átadja neki v-t. Ezután meghívja a v getBearDancingjét, ha ez igazzal tér vissza akkor beállítja az Empty attribútumot igazra, az AddSteppable-nek átadva magát bekerül a léptethető objektumok közé és beállítja a remaningTime attribútumot 2-re. Egyébként meghívja a wannaPickUp metódust. Ha ez igazzal tért vissza és az Empty hamis, akkor meghívja az addMat metódust és átadja neki v-t. **Láthatóság:** +
* **void addMat(Virologist v)**: Generál egy random egész számot 0 és 1 között. Majd egy switch case szerkezetben a 2 esetet szétválasztva más és más típusú anyagot hoz létre és meghívja az adott anyagnak megfelelő pickUp metódust átadva neki v-t. (0 esetén aminosavat hoz létre, 1 esetén nukleotidot és így tovább). **Láthatóság:** -
* **boolean wannaPickUp()**: Megkérdezi a játékost, hogy akar-e anyagot felvenni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **void step()**: Csökkenti a remaningTime-ot egyel. Ha a remaningTime nulla, akkor beállítja az Empty attribútumot hamisra, majd meghívja a Timer RemoveSteppable függvényét és átadja önmagát.

### Shelter

#### Felelősség

Speciális típusú mező, amelyen eszközöket szedhet fel az ide érkező virológus.

#### Ősosztályok

Field

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát és átadja neki v-t. Majd meghívja a wannaPickUp metódust, ha ez igazzal tér vissza, akkor meghívja az addEq metódust és átadja neki v-t. **Láthatóság:** +
* **void addEq(Virologist v)**: Generál egy random egész számot 0 és 3 között és egy switch case szerkezetben a 4 esetet szétválasztva más és más típusú eszközt hoz létre és meghívja az adott eszköznek megfelelő pickUp metódust átadva neki v-t.. (pl. 0 esetén zsákot hoz létre, 1 esetén kesztyűt és így tovább). **Láthatóság:** -
* **boolean wannaPickUp()**: Megkérdezi a játékost, hogy akar-e eszközt felvenni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -

### Lab

#### Felelősség

Speciális típusú mező, amelyen genetikai kódokat ismerhet meg az ide érkező virológus.

#### Ősosztályok

Field

#### Interfészek

Nincs

#### Attribútumok

* **GCode gc**: Tárolja a laborban megtanulható genetikai kód objektumot. **Láthatóság:** -

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát és átadja neki v-t. Majd meghívja a wannaTouchy metódust, ha ez igazzal tér vissza, akkor meghívja a Virológus touchy metódusát és átadja neki gc-t. **Láthatóság:** +
* **boolean wannaPickUp()**: Megkérdezi a játékost, hogy akar-e genetikai kódot letapogatni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -

### InfectedLab

#### Felelősség

Speciális típusú labor, amelyen a virológus megfertőződhet medvetánccal.

#### Ősosztályok

Field → Lab

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void arrive(Virologist v)**: Először meghívja az ősosztály arrive metódusát. A v getImmunity metódus hívás megadja, hogy a virológus immunis-e. Ha immunis akkor visszatér az arrive. Ezután a v getWornCloak metódussal elkéri a v által viselt köpenyek listáját. Ha ezen listának a hossza nagyobb, mint nulla, akkor egy ciklussal végig megy a kapott listán és minden Cloak objektum evade-jét meghívja. Ha az egyik evade igazzal tér vissza akkor törli az objektumot a listából és visszatér az arrive. Majd meghívja a v setBearDancing metódusát true értékkel, létrehoz egy új BearDance objektumot és a Timer AddSteppable-jének átadva v-t bekerül a léptethető dolgok közé. **Láthatóság:** +

### Agent

#### Felelősség

Az elkészíthető ágensek ősosztálya. Tárolja az ágens nevét, azon virológust akin hat, az ágens hatásának időtartamát, az időtartamot, ameddig felhasználható az ágens és fennmaradó időt. Az ágensek hatásait kezeli.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **int RemaningTime**: Tárolja, hogy hány kör után fog megszűnni az ágens. **Láthatóság:** -
* **String Name**: Tárolja az ágens nevét. **Láthatóság:** -
* **int Expiration**: Tárolja, hogy hány körig használható fel az ágens. **Láthatóság:** -
* **int EffectTime**: Tárolja, hogy hány körig hat a felkent ágens. **Láthatóság:** -
* **Virologist VirologistUnderEffect**: Tárolja, hogy melyik virológusra hat az ágens. **Láthatóság:** -

#### Metódusok

* **void step()**: absztrakt metódus. **Láthatóság:** +
* **void setVirologistUnderEffect(Virologist v)**: Beállítja a VirologistUnderEffectattribútumot v-re. **Láthatóság:** +
* **void getEffectTime()**: Visszaadja az EffectTime attribútum értékét. **Láthatóság:** +
* **void setRemaningTime(int time)**: Beállítja a RemaningTime attribútum értékét time-ra. **Láthatóság:** +
* **void getRemaningTime()**: Visszaadja a RemaningTime attribútum értékét. **Láthatóság:** +

### Oblivion

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus elfelejti az összes genetikai kódot amit eddig megtanult.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: Ha a VirologistUnderEffect nem null, akkor meghívja VirologistUnderEffect getInventory függvényét, majd az Inventory forgetGCodes metódusát, beállítja a RemaningTime-t nullára és meghívja a Timer RemoveSteppable függvényét és átadja önmagát. **Láthatóság:** +

### Paralysis

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus lebénul, így nem tud semmit csinálni.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: Ha a VirologistUnderEffect nem null, akkor meghívja VirologistUnderEffect setParalyzed függvényét igaz értéket átadva, csökkenti a RemaningTime-t egyel. Ha a RemaningTime nulla, akkor meghívja VirologistUnderEffect setParalyzed függvényét hamis értékkel és a Timer RemoveSteppable függvényét és átadja önmagát. **Láthatóság:** +

### Immunity

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus immunissá válik, tehát nem hatnak rá a mások által rákent ágensek.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: Ha a VirologistUnderEffect nem null, akkor meghívja VirologistUnderEffect setImmunity függvényét igaz értéket átadva, csökkenti a RemaningTime-t egyel. Ha a RemaningTime nulla, akkor meghívja VirologistUnderEffect setImmunity függvényét hamis értékkel és a Timer RemoveSteppable függvényét és átadja önmagát. **Láthatóság:** +

### VitusDance

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus lebénul, így nem tud semmit csinálni.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**:

Ha: VirologistUnderEffect != null, akkor

VirologistUnderEffect setDancing(igaz)

f ← VirologistUnderEffect getStandingField

Neighbours ← f getNeighbours()

chosenField ← Neighbours lista egyik eleme

chosenField move(VirologistUnderEffect,hamis)

Ha: RemaningTime = nulla, akkor

VirologistUnderEffect setDancing(hamis)

Time RemoveSteppable(ez a VitusDance objektum)

**Láthatóság:** +

### BearDance

#### Felelősség

Speciális típusú ágens, melynek hatására a virológus minden vele egy mezőn tartózkodó virológust megtámad BearDance ágenssel.

#### Ősosztályok

Agent

#### Interfészek

Steppable

#### Attribútumok

Nincs

#### Metódusok

* **void step()**:

Ha: VirologistUnderEffect != null, akkor

VirologistUnderEffect setBearDancing(igaz)

f ← VirologistUnderEffect getStandingField

infect(f)

Ha: VirologistUnderEffect getDied() = hamis, akkor

Neighbours ← f getNeighbours()

chosenField ← Neighbours lista egyik eleme

chosenField move(VirologistUnderEffect,hamis)

infect()

Ha: VirologistUnderEffect getDied() = igaz, akkor

Time RemoveSteppable(ez a BearDance objektum)

**Láthatóság:** +

* **void infect(f)**:

virologists ← f getVirologists()

Ciklus: amíg i kisebb, mint virologist lista hossza

Ha: VirologistUnderEffect != i-dik Virologist objektum, akkor

i-dik Virologist attack(ez a BearDance objektum)

Ciklus vége

**Láthatóság:** -

### Steppable

#### Felelősség

Egy interface, amely a léptethető objektumokat kezeli.

#### Ősosztályok

Nincs

#### Interfészek

Nincs

#### Attribútumok

Nincs

#### Metódusok

* **void step()**: üres metódus. **Láthatóság:** +

### Game

#### Felelősség

Létrehozza, inicializálja a mezőket, virológusokat és felépíti a pályát. Kezeli, hogy melyik virológus cselekedhet éppen és egy idő után lép a következő virológusra ezt a játék végig csinálja. Ha vége a játéknak minden virológust értesít erről majd lebontja a pályát.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **int MaxGCode**: Tárolja, hogy hány különböző genetikai kódot kell összegyűjtenie a virológusoknak, csak olvasható. **Láthatóság:** -
* **int activeVirologist**: Tárolja, hogy a virológusok lista hányadik eleme az éppen aktív virológus. **Láthatóság:** -
* **boolean moved**: Tárolja, hogy a soron következő virológus mozgot-e már ebben a körben, hamisra van inicializálva. **Láthatóság:** -
* **ArrayList** **<Field> fields**: Tárolja a játék összes mezőjét. **Láthatóság:** -
* **ArrayList** **<Virologist> virologists**: Tárolja a játékban resztvevő virológusokat. **Láthatóság:** -
* **Timer timer**: Tárolja, a játék során használt időzítő objektumot. **Láthatóság: -**
* **long startTime**: Tárolja, hogy a virológus mikor kezdte a körét. **Láthatóság:** -
* **int roundTime**: Tárolja, hogy a virológus köre maximum meddig tart. **Láthatóság:** -

#### Metódusok

* **int getMoved()**: Visszaadja a moved attribútum értékét. **Láthatóság:** +
* **void setMoved(boolean moved)**: Beállítja a moved attribútumot a kapott értékre. **Láthatóság:** +
* **Virologist getActiveVirologist()**: Visszaadja a virologists lista azon elemét, amely activeVirologist sorszámú helyen van. **Láthatóság:** +
* **void removeVirologist(Virologist v)**: Eltávolítja a kapott Virologist objektumot a játékban levő virológusok listájából **Láthatóság:** +
* **boolean doCommand()**: Visszatér, hogy tud-e cselekedni/cselekedett-e valamit az éppen körre kerülő játékos vagy sem. **Láthatóság:** -
* **void startGame()**: Felépíti a mezőket, magát a pályát. Létrehozza a virológusokat, azokat elhelyezi kezdeti mezőjükön, majd elindítja a játékot, azaz az első kört. **Láthatóság:** +
* **void endGame(Virologist v)**:

Ha v nem null, akkor

ciklus: amíg i kisebb, mint fields lista hossza

ciklus: amíg j kisebb, mint a virologists lista hossza

meghívja az i-dik Field j-dik Virologist objektum showEndGame-t v-vel

ciklus vége

ciklus vége

**Láthatóság:** +

* **void step()**: Ha az activeVirologist = nulla, akkor:

ciklus: s végig fut Steppables léptethető objektumokon

meghívja s step metódusát

ciklus vége

next ←hamis

startTime ← rendszer idő

ciklus: amíg rendszer idő – startTime < roundTime és !next

next ← doCommand

ciklus vége

activeVirologist ← 1+activeVirologist mod virologists lista hossza

move ← hamis

**Láthatóság:** +

### Virologist

#### Felelősség

Lehetővé teszi a felhasználó és a rendszer közötti kommunikációt. A játékos ezen az osztályon keresztül mozoghat a szomszédos mezők között, ágenst generálhat, felvehet és levehet eszközöket (amelyekből maximum 3 lehet rajta egyszerre), ellenfeleire vagy önmagára ágenst kenhet, és védekezhet egy általa indított támadás (ágens kenés) ellen. Emellett eszközöket, anyagokat lophat egy bénult virológustól.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **boolean Immunity**: Tárolja, hogy a virológus immunis-e. **Láthatóság:** -
* **boolean Paralyzed**: Tárolja, hogy a virológus le van-e bénulva. **Láthatóság:** -
* **boolean Dancing**: Tárolja, hogy a virológus épp vitustánc hatása alatt áll-e. **Láthatóság:** -
* **boolean BearDancing**: Tárolja, hogy a virológus épp medvetánc hatása alatt áll-e. **Láthatóság:** -
* **boolean Dead**: Tárolja, hogy a virológus meghalt-e. **Láthatóság:** -
* **ArrayList** **<Gloves> WornGloves**: A virológus által viselt kesztyűk listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **ArrayList** **<Sack> WornSacks**: A virológus által viselt zsákok listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **ArrayList** **<Cloak> WornCloaks**: A virológus által viselt köpenyek listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **ArrayList** **<Axe> WornAxes**: A virológus által viselt fejszék listája, melyben minimum 0, maximum 3 elem lehet (attól függően, hogy a másik három eszközfajta listája mennyi elemből áll). **Láthatóság:** -
* **Field standingField**: Az a mező, amelyen a virológus áll. **Láthatóság:** -
* **Inventory inventory**: Tárolja a virológus inventoryját. **Láthatóság:** -
* **Agent agents**: A virológus által készített ágensek listája. **Láthatóság:** -
* **Game game**: Az a játék, amelyen a virológus részt vesz. **Láthatóság:** -

#### Metódusok

* **void setImmunity(boolean bImmun)**: Beállítja az Immunity attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setParalyzed(boolean bParalyzed)**: Beállítja a Paralyzed attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setDancing(boolean bDancing)**: Beállítja a Dancing attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setBearDancing(boolean bBearDancing)**: Beállítja az BearDancing attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void setStandingField(Field f)**: Beállítja a standingField attribútum értékét a bementi paraméterére. **Láthatóság:** +
* **void die()**: Beállítja az Died attribútum értékét igazra. **Láthatóság:** +
* **boolean getImmunity()**: Visszaadja az Immunity attribútum értékét. **Láthatóság:** +
* **boolean getParalyzed()**: Visszaadja a Paralyzed attribútum értékét. **Láthatóság:** +
* **boolean getDancing()**: Visszaadja a Dancing attribútum értékét. **Láthatóság:** +
* **boolean getBearDancing()**: Visszaadja a BearDancing attribútum értékét. **Láthatóság:** +
* **Field getStandingField()**: Visszaadja a standingField attribútum értékét.  
  **Láthatóság:** +
* **Inventory getInventory()**: Visszaadja az inventory attribútum értékét. **Láthatóság:** +
* **void init(Game game, Field standingField)**: Ha a paraméterek nem null értékűek, akkor beállítja a game és standingField attribútumok értékeit a bementi paraméterekre,. Egyébként létrehoz új objektumokat. Majd beállítja az Immunity, Paralyzed, Dancing, BearDancing és Died attribútumokat hamisra. Az agents, WornGloves, WornSacks, WornCloaks, WornAxes és az inventory attribútumokat inicializálja. **Láthatóság:** +
* **void step()**:Ciklus: amíg i kisebb mint az agents lista hossza

a ← i-dik Agent

meghívja a setRemaningTime-ját a getRemaningTime-1 -el

ha: a getRemaningTime-ja = 0, akkor

agents-ből törli az i-dik elemet

különben:

agents i-dik elemét a-ra állítja

ciklus vége

**Láthatóság:** +

* **void equipGloves()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeGloves metódusát, amely egy Gloves objektummal tér vissza, ha ez nem null, akkor a WornGloves listához hozzáadja ezt. **Láthatóság:** +
* **void equipSack()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeSack metódusát amely egy Sack objektummal tér vissza, ha ez nem null, akkor a WornSacks listához hozzáadja ezt, elkéri az inventory-tól, hogy mennyi a jelenlegi maxMaterial (currentMaxMaterial ← inventory getMaxMaterial) és az inventory setMaxMaterialnak átadja az elkért MaxMaterial és az adott zsák kapacitásának összegét (setMaxMaterial(currentMaxMaterial+sack getCapacity())). **Láthatóság:** +
* **void equipCloak()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeCloak metódusát, amely egy Cloak objektummal tér vissza, ha ez nem null, akkor a WornCloaks listához hozzáadja ezt. **Láthatóság:** +
* **void equipAxe()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha a canEquipEq igazat ad vissza, akkor meghívja az inventory removeAxe metódusát, amely egy Axe objektummal tér vissza, ha ez nem null, akkor a WornAxes listához hozzáadja ezt. **Láthatóság:** +
* **boolean defense(Virologist vFr, Agent a)**:

bDefenseSucceed ← hamis

Ha: Dancing = igaz vagy Paralyzed = igaz, vagy BearDancing = igaz, akkor

Vissza: bDefenseSucceed

Különben, ha: a = null vagy vFr = null, akkor

bDefenseSucceed ← igaz

Vissza: bDefenseSucceed

Különben:

Ha: WornGloves lista hossza > 0, akkor

g ← WornGloves 0-dik eleme

Ha: g wannaUse() = igaz, akkor

usable ← g getUsable()

g setUsable(usable-1)

Ha usable = 1, akkor

törli a WornGloves 0-dik elemét

Különben:

g use(vFr, önmaga, a)

bDefenseSucceed ← igaz

Ha bDefenseSucceed = hamis, akkor

Ha WornAxes lista hossza > 0, akkor

a ← WornAxes 0-dik eleme

used ← a getUsed()

Ha: a wannaUse() = igaz és used = hamis, akkor

törli a WornAxes 0-dik elemét

a use(vFr)

bDefenseSucceed ← igaz

Ha bDefenseSucceed = hamis, akkor

ciklus: amíg i kisebb, mint az WornCloak lista hossza

c ← WornCloaks i-dik eleme

bDefenseSucceed ← c evade()

Ha bDefenseSucceed = igaz, akkor

törli a WornCloaks i-dik elemét

stop

ciklus vége

Vissza: bDefenseSucceed

**Láthatóság:** +

* **void takeOffGloves()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként meghívja a takeOffGloves metódust átadva neki az inventoryt. **Láthatóság:** +
* **boolean takeOffGloves(Inventory inv)**:

bFound ← hamis

Ha WornGloves lista hossza > 0, akkor

g ← WornGloves 0-dik eleme

inv addGloves(g)

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void takeOffCloak()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként meghívja a takeOffCloak metódust átadva neki az inventoryt. **Láthatóság:** +
* **boolean takeOffCloak(Inventory inv)**:

bFound ← hamis

Ha WornCloaks lista hossza > 0, akkor

g ← WornCloaks 0-dik eleme

inv addCloak(g)

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void takeOffAxes()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként, ha: a wannaDropAxe hamissal tér vissza, akkor meghívja a takeOffAxes metódust átadva neki az inventoryt. Ha wannaDropAxe igazzal tér vissza, akkor törli a WornAxes 0-dik elemét. **Láthatóság:** +
* **boolean takeOffAxes(Inventory inv)**:

bFound ← hamis

Ha WornAxes lista hossza > 0, akkor

g ← WornAxes 0-dik eleme

inv addAxes(g)

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void takeOffSack()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként meghívja a takeOffSack metódust átadva neki az inventoryt. **Láthatóság:** +
* **boolean takeOffSack(Inventory inv)**:

bFound ← hamis

Ha WornSacks lista hossza > 0, akkor

s ← WornSacks 0-dik eleme

inv addSack(s)

currentMaxMaterial ← inv getMaxMaterial

inv setMaxMaterial(currentMaxMaterial + s getCapacity())

bFound ← igaz

Vissza: bFound

**Láthatóság:** -

* **void createAgent(GCode gc)**:

Ha: a virológus lebénult vagy vitustáncol vagy medvetáncol, akkor

Vissza:

Ha: gc != null, akkor

a ← gc create (önmaga)

Ha: a != null, akkor

agents listához hozzáadja a-t

**Láthatóság:** +

* **void attack(Virologist vTo, Agent a)**:

Ha: Dancing = igaz vagy Paralyzed = igaz vagy Immunity = igaz vagy standingField != vTo standingField, akkor

Vissza:

Ha vTo != null és a != null, akkor

bDefenseSuccessful = vTo assaulted(önmaga, a)

Ha: bDefenseSuccessful != hamis és vTo getParalyzed(), akkor

Ha wannaSteal(), akkor

vTo stealBy(önmaga)

**Láthatóság: +**

* **boolean wannaSteal()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként megkérdezi a játékost, hogy akar-e lopni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **void move(Field f, boolean bEmittedByUser)**:

Ha: Dancing = igaz vagy Paralyzed = igaz, vagy standingField szomszédai között nincsen f, akkor

Vissza:

Ha: f != null, akkor

standingField RemoveVirologist(önmage)

f addVirologist(önmaga)

standingField ← f

Ha: bEmittedByUser, akkor

f arrive(önmaga)

**Láthatóság:** +

* **void stealBy(Virologist v)**:

Ha: v = null vagy v Paralyzed = igaz, vagy v Dancing = igaz, vagy v BearDancing = igaz, akkor

Vissza:

Ha: Paralyzed = igaz, akkor,

stealerInv ← v getInventory()

objectName ← v chooseObjectToSteal

bEquippedFound ← hamis

switch(objectName)

case Aminoacid: stealerInv addAmino(inventory removeAmino)

case Nukleotid: stealerInv addNukleo(inventory removeNukleo)

case Gloves:

bEquippedFound ← takeOffGloves(inventory removeGloves())

Ha bEquippedFound =hamis, akkor

stealerInv addGloves(inventory removeGloves)

case Sack:

bEquippedFound ← takeOffSack(inventory removeSack())

Ha bEquippedFound =hamis, akkor

stealerInv addSack(inventory removeSack)

case Cloak:

bEquippedFound ← takeOffGloves(inventory removeCloak())

Ha bEquippedFound =hamis, akkor

stealerInv addCloak(inventory removeCloak)

case Axe:

bEquippedFound ← takeOffAxe(inventory removeAxe())

Ha bEquippedFound =hamis, akkor

stealerInv addAxe(inventory removeAxe) **Láthatóság:** +

* **boolean assaulted(Virologist vFr, Agent a)**:

bDefenseSucceed ← hamis

Ha: getHasEffectedAgent() = hamis és vFr != null és a != null, akkor

Ha: vFr = ez a Virologist objektum, akkor

bDefenseSucceed ← defense(vFr, a)

Ha: bDefenseSucceed = hamis, akkor

a setVirologistUnderEffect(ez a Virologist objektum)

a setRemaningTime(a getEffectTime())

Timer addSteppable(a)

Vissza: bDefenseSucceed

**Láthatóság:** -

* **String chooseObjectToSteal()**: Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként megkérdezi a játékost, hogy milyen anyagot vagy eszközt akar lopni. És visszatér a felhasználó által beadott Stringel. **Láthatóság:** -
* **boolean canEquipEq()**: Ha a WornGloves, WornSacks, WornCloaks, WornAxes listák hosszainak összege kisebb, mint 3, akkor igazzal tér vissza. Ha nagyobb vagy egyenlő akkor hamissal. **Láthatóság:** -
* **boolean getHasEffectedAgent()**: Visszaaadja az Immunity és Dancing és Paralyzed logikai kifejezés értékét. **Láthatóság:** -
* **void showEndGame(Virologist v)**:

Ha: v != null és v != ez a Virologist objektum, akkor

Kiír: Game over! You Lost.

Különben:

Kiír: Congratulation! You won the game.

* **boolean wannaDropAxe()**:Ha a virológus lebénult vagy vitustáncol vagy medvetáncol ,akkor nem történik semmi. Egyébként megkérdezi a játékost, hogy akar-e fejszét használni. Ha a válasz y/Y akkor igazzal tér vissza, ha a válasz n/N akkor hamissal tér vissza. **Láthatóság:** -
* **ArrayList** **<Cloak> getWornCloaks()**: Visszaadja a virologús által viselt köpenyek listáját. **Láthatóság:** +
* **void die()**: Beállítja a Died attribútumot igazra. **Láthatóság:** +

### Timer

#### Felelősség

Periodikus időt reprezentál a játékban, a léptethető (Steppable) dolgokat lépteti.

#### Ősosztályok

Nincs

#### Interfészek

Steppable

#### Attribútumok

* **Steppable Steppables**: Tárolja a léptethető dolgokat. **Láthatóság:** -

#### Metódusok

* **void tick()**: Minden léptethető dolgot léptet az előre beállított értékkel. **Láthatóság:** +
* **void addSteppable(Steppable s)**: Új léptethető dolog felvétel a Steppable listára. **Láthatóság:** +
* **void removeSteppable(Steppable s)**: A lejárt idejű léptető (tehát a léptető elérte a nullát) törlése kerül ezáltal a metódus által a Steppable listából. **Láthatóság:** +

## A tesztek részletes tervei, leírásuk a teszt nyelvén

### Teszteset1:

#### Leírás

Az óvóhelyen megszerezhető eszközöket reprezentáló absztrakt osztály.

## A tesztelést támogató programok tervei

### Egy teszt lefutása

A tesztelés úgy fog lezajlani, hogy a program tartalmazni fog egy külön részt, amely kifejezetten a tesztekkel foglalkozik, így indítást követően az elindul, a megadott sorrendbe beolvassa a tesztfájlokat, meg beadja az első teszthez a megadott bemenetet, és elkészíti a kimenetet hozzá.

### Tesztek lefutásához program

A tesztekhez JUnit könyvtárat fogunk használni, ugyanis az lett számunkra a legkézenfekvőbb és nem kell különösebben új programot feltelepíteni, engedélyezni dolgokat a gépen.

### Elvárt és tényleges kimenet összehasonlítása

A tesztfájlok beolvasása azért szükséges, mert azokban vannak tárolva az elvárt kimenetek, így, mikor a program legenerálja azokat, össze tudjuk hasonlítani az elvárt (tesztfájlos) kimenetet a kigeneráltal. Ha a kettő azonos, akkor sikeres volt a tesztünk, ha eltérés mutatkozik, akkor meg sajnos hibára futottunk.

## Ütemterv

|  |  |  |  |
| --- | --- | --- | --- |
| **Határidő** | **Feladat** | **Pont** | **Felelős** |
| **febr. 28.** | Követelmény, projekt, funkcionalitás - beadás | 10 | Csia |
| **márc. 7.** | Analízis modell kidolgozása I. - beadás | 20 | Alpek |
| **márc. 16.** | Analízis modell kidolgozása II. - beadás | 30 | Marton |
| **márc. 21.** | Szkeleton tervezése - beadás | 20 | Litavecz |
| **márc. 28.** | Szkeleton elkészítése - beadás | 20 | Csia |
| **ápr. 4.** | Prototípus koncepciója – beadás  Szkeleton bemutatás | 20 | Alpek |
| **ápr. 11.** | Részletes tervek - beadás | 45 | Marton |
| **ápr. 25.** | Prototípus készítése, tesztelése | 35 |  |
| **máj. 2.** | Grafikus változat tervei – beadás  Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése | 30 |  |
| **máj. 16.** | Grafikus változat készítése | 40 |  |
| **máj. 18.** | Egységes dokumentáció - beadás és bemutatás  Grafikus változat - beadás és a forráskód herculesre való feltöltése, és teljes házi bemutatás | 30 |  |

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevő(k)** | **Leírás** |

|  |  |  |  |
| --- | --- | --- | --- |
| 2022.04.07 20:30 | 5 óra | Alpek, Litavecz, Marton, Ruskó | Értekezlet: Új feladat átbeszélése, tervek elkészítése a feladat teljesítéséhez. |
| 2022.04.09 12:00 | 7 óra | Litavecz | Feladat: Az osztályok és metódusok rész megírása. |
| 2022.04.11 18:00 | 4,5 óra | Alpek, Csia, Marton, Ruskó | Értekezlet: Bemeneti nyelv alapjainak leszögezése, tesztek megírásához sablon elkészítése, feladatok szétosztása egymás közt. Csia: grafikus felülettel már előre kalkulálás, és azzal tervezni mostani kódhoz. |
| 2022.04.12. 00:00 | 6 óra | Csia | Feladat: Dokumentum összeállítása, Litavecz feladatrészének átnézése, maradék metódusrész kibővítése, hiányzó osztály elhelyezése. A 8.3-as rész és a Napló megírása. |
| 2022.04.12. 07:00 | 7 óra | Alpek | Feladat: Tesztesetek elkészítése a hozzájuk tartozó leírásokkal, illetve bemeneti/kimeneti részekkel. (Fejenként 29 db.) |
| 2022.04.12. 03:00 | 7 óra | Marton | Feladat: Tesztesetek elkészítése a hozzájuk tartozó leírásokkal, illetve bemeneti/kimeneti részekkel. (Fejenként 29 db.) |
| 2022.04.12. 06:00 | 7 óra | Ruskó | Feladat: Tesztesetek elkészítése a hozzájuk tartozó leírásokkal, illetve bemeneti/kimeneti részekkel. (Fejenként 29 db.) |
| 2022.04.12. 11:00 | 1 óra | Csia | Feladat: A dokumentum összeillesztése a maradék munkákkal, egységesítése, a dokumentum felöltése, leadása. |

## Százalékos teljesítés

|  |  |
| --- | --- |
| **Név** | **Százalék** |

|  |  |
| --- | --- |
| Alpek | 20% |
| Csia | 20% |
| Litavecz | 20% |
| Marton | 20% |
| Ruskó | 20% |