Drilling

#### Felelősség

Ez az interfészt azok az osztályok valósítják meg, amik tudnak fúrni.

#### Állapotdiagram

-

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

#### -

#### Metódusok

* **+drill():** Egy absztrakt függvény amely elvégzi azokat a fúrással kapcsolatos dolgokat, amiket a fúrásra képes osztályoknak kell

Mining

#### Felelősség

Ez az interfészt azok az osztályok valósítják meg, amik képesek nyersanyagot kibányászni.

#### Állapotdiagram

-

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

#### Metódusok

* **+mine():** egy absztrakt függvény, amely elvégzi a azokat a bányászással kapcsolatos dolgokat, amiket a bányászára képes osztályoknak kell.

Moving

#### Felelősség

Ez az interfészt azok az osztályok valósítják meg, amik képesek mozogni.

#### Állapotdiagram

-

#### Ősosztályok

-

#### Interfészek

-

#### Attribútumok

#### Metódusok

* **+move(destination: Asteroid):** egy absztrakt függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol a cél egy másik aszteroida, amiket a mozgásra képes osztályoknak kell.
* **+move(destinationGate: StarGate):** egy absztrakt függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol egy teleport kaput használunk, amiket a mozgásra képes osztályoknak kell.

Entity

#### Felelősség

Ez az absztrakt ősosztály felel azokért, akik meg tudnak halni, és érinti őket valahogy az aszteroida felrobbanása.

#### Állapotdiagram

-

#### Ősosztályok

Printable->Entity

#### Interfészek

ReactsToSunflare, Moving

#### Attribútumok

* **-id: int:** Ez az adott Entity id-je, ezt arra használjuk fel, hogy az entity-t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott entitiy fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén. Az Entitiy osztályban ez egy absztakt függvény
* **+getNotifiedAboutSunflare():** Ez a függvény akkor hívódik meg az entity-n amikor azt napvihar éri, ilyenkor az reagál arra ha kell neki.
* **+move(destination: Asteroid):** egy absztrakt függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol a cél egy másik aszteroida, amiket a mozgásra képes osztályoknak kell.
* **+move(destinationGate: StarGate):** egy absztrakt függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol egy teleport kaput használunk, amiket a mozgásra képes osztályoknak kell.
* **+die():** ez az absztrakt függvény végzi el a meghalással kapcsolatos dolgokat. Ez az Entity-ben absztrakt.
* **+asteroidExploded():** ez az absztrakt függvény biztosítja, hogy hogyan viselkedjen az osztály, amikor alatta egy aszteroida felrobban.

NonPlayer

#### Felelősség

Ez az absztrakt ősosztály felel a nem játékos álltál irányítót karakterekért

#### Ősosztályok

Printable->Entity->NonPlayer

#### Interfészek

ReactsToSunflare, Moving,Controllable

#### Attribútumok

* **-id: int:** Ez az adott NonPLayer id-je, ezt arra használjuk fel, hogy az NonPLayer-t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* -**asteroid:Asteroid**: Ezen az aszteroidán van a NonPlayer.
* -**owner:Game**: Ez az osztály tárolja le a NonPlayer-t

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott NonPlayer fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén. Az NonPlayer osztályban ez egy absztrakt függvény
* **+getNotifiedAboutSunflare():** Ez a függvény az osztály die függvényét hívja, amennyiben az asteroida-ja nem igaz értékkel tér vissza az isHollow függvényre
* **+move(destination: Asteroid):** egy függvény, aszteroidáról aszteroidára való mozgást végzi.

|  |
| --- |
| If(destination is reachable from NonPlayer.asteroid) {  destination.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid=destination } |

* **+move(destinationGate: StarGate):** ez a függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol egy teleport kaput használunk, amiket a mozgásra képes osztályoknak kell.

|  |
| --- |
| If(destinationGate is active) {  destinationGate.neighbour.asteroid.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid= destinationGate.neighbour.asteroid } |

* **+die():** ez a függvény kiveszi a NonPlayer-t az aszteroidáról, és jelzi a játéknak, hogy kivehető a létező elemekből.
* **+asteroidExploded():** ez a függvény biztosítja, hogy hogyan viselkedjen az osztály, amikor alatta egy aszteroida felrobban.
* **+onTurn():** Egy absztrakt függvény mely elvégzi az adott objektum egy körben elvégzendő feladatait.

Alien

#### Felelősség

Ez az osztály felel az űrlényért és függvényeiért. Pl.: hogy az ürlény vegyen fel nyersanyagot

#### Ősosztályok

Printable->Entity->NonPlayer->Alien

#### Interfészek

ReactsToSunflare, Moving, Mining

#### Attribútumok

* **-id: int:** Ez az adott Alien id-je, ezt arra használjuk fel, hogy az Alien-t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* -**asteroid:Asteroid**: Ezen az aszteroidán van az Alien.
* -**owner:Game**: Ez az osztály tárolja le az Alien-t
* **-inventory:Inventory:** Ebben tárolja az Alien általa össze gyűjtőt nyersanyagokat.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Alien fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+getNotifiedAboutSunflare():** Ez a függvény az osztály die függvényét hívja, amennyiben az asteroida-ja nem igaz értékkel tér vissza az isHollow függvényre
* **+move(destination: Asteroid):** egy függvény, aszteroidáról aszteroidára való mozgást végzi.

|  |
| --- |
| If(destination is reachable from NonPlayer.asteroid) {  destination.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid=destination } |

* **+move(destinationGate: StarGate):** ez a függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol egy teleport kapút használunk, amiket a mozgásra képes osztályoknak kell.

|  |
| --- |
| If(destinationGate is active) {  destinationGate.neighbour.asteroid.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid= destinationGate.neighbour.asteroid } |

* **+die():** ez a függvény kiveszi a Alien-t az aszteroidáról, és jelzi a játéknak, hogy kivehető a létező elemekből.
* **+asteroidExploded():** ez a függvény meghívja az Alien die függvényét.
* **+onTurn():** Egy függvény mely elvégzi az Alien objektum egy körben elvégzendő feladatait. Pl.: mozgás vagy bányászás.
* **+mine()**: egy függvény, amely kivesz egy Material-t az aszteroidából, ha az aszteroida kérge 0.

Robot

#### Felelősség

Ez az osztály felel az robotértés függvényeiért.

#### Ősosztályok

Printable->Entity->NonPlayer->Robot

#### Interfészek

ReactsToSunflare, Moving, Mining

#### Attribútumok

* **-id: int:** Ez az adott Robot id-je, ezt arra használjuk fel, hogy az Robot-t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* -**asteroid:Asteroid**: Ezen az aszteroidán van a Robot.
* -**owner:Game**: Ez az osztály tárolja le a Robot-t
* **-isRandom: Boolean:** Egy boolean ami azt jelzi, hogy a Sun random módon, vagy determinisztikusan működik-e (kitörés esély számlálás közben), ez csak a teszteléshez kell.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Robot fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+getNotifiedAboutSunflare():** Ez a függvény az osztály die függvényét hívja, amennyiben az asteroida-ja nem igaz értékkel tér vissza az isHollow függvényre
* **+move(destination: Asteroid):** egy függvény, aszteroidáról aszteroidára való mozgást végzi.

|  |
| --- |
| If(destination is reachable from NonPlayer.asteroid) {  destination.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid=destination } |

* **+move(destinationGate: StarGate):** ez a függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol egy teleport kaput használunk, amiket a mozgásra képes osztályoknak kell.

|  |
| --- |
| If(destinationGate is active) {  destinationGate.neighbour.asteroid.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid= destinationGate.neighbour.asteroid } |

* **+die():** ez a függvény kiveszi a Robo-t az aszteroidáról, és jelzi a játéknak, hogy kivehető a létező elemekből.
* **+asteroidExploded():** ez a függvény biztosítja, hogy hogyan viselkedjen az osztály, amikor alatta egy aszteroida felrobban. A robot esetében ez move-t hív a robot aszteroidjának egyi szomszédjával mint paraméter. Amennyiben az nem lehetséges Robot die függvényét hívja.
* **+onTurn():** Egy függvény mely elvégzi a Robot objektum egy körben elvégzendő feladatait. Pl.: mozgás vagy fúrás.
* **+drill():** Ez a függvény meghívja a Robot asteroid-n a drilling függvényt.

Settler

#### Felelősség

Ez az osztály felel a játékos által irányítható telepesért.

#### Ősosztályok

Printable->Entity->Settler

#### Interfészek

ReactsToSunflare, Moving, Drilling, Mining

#### Attribútumok

* **-id: int:** Ez az adott Settler id-je, ezt arra használjuk fel, hogy az Settler-t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* -**asteroid:Asteroid**: Ezen az aszteroidán van a Settler.
* -**owner:Game**: Ez az osztály tárolja le a Robot-t
* **-inventory:Inventory:** Ebben tárolja a Settler általa össze gyűjtőt nyersanyagokat.
* **-hadActionThisTurn:Boolean**: Ez tárolja, hogy lehet e ezzel a telepessel lépni az adott körben.
* **-baseRecipe**: ebben van benne, hogy mi kell egy bázis megépítéséhez.
* **- robotRecipe**: ebben van benne, hogy mi kell egy robot megépítéséhez.
* **-stargateRecipe**: ebben van benne, hogy mi kell egy kapú megépítéséhez.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Settler fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+getNotifiedAboutSunflare():** Ez a függvény az osztály die függvényét hívja, amennyiben az asteroida-ja nem igaz értékkel tér vissza az isHollow függvényre
* **+move(destination: Asteroid):** egy függvény, aszteroidáról aszteroidára való mozgást végzi.

|  |
| --- |
| If(destination is reachable from NonPlayer.asteroid) {  destination.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid=destination } |

* **+move(destinationGate: StarGate):** ez a függvény, amely elvégzi azokat a mozgással kapcsolatos dolgokat ahol egy teleport kaput használunk, amiket a mozgásra képes osztályoknak kell.

|  |
| --- |
| If(destinationGate is active) {  destinationGate.neighbour.asteroid.add(NonPlayer)  NonPlayer.asteroid.remove(NonPlayer)  NonPlayer.asteroid= destinationGate.neighbour.asteroid } |

* **+die():** ez a függvény kiveszi a Settler-t az aszteroidáról, és jelzi a játéknak, hogy kivehető a létező elemekből.
* **+asteroidExploded():** ez a függvény meghívja a Settler die függvényét.
* **+drill():** Ez a függvény meghívja a Settler asteroid-n a drilling függvényt.
* **+mine()**: egy függvény, amely kivesz egy Material-t az aszteroidából, ha az aszteroida kérge 0.
* **+buildStarGate():** Amennyiben van kellő nyers anyag azokat kiveszi az Inventory-ból és egy kapu párthelyez az Inventory-ba
* **+buildRobot():** Amennyiben van kellő nyersanyag létrehoz egy robotot az aszteroidán, amin tartozkódik, és kiveszi a felhasznált nyersanyagokat az Inventory-ból
* **+buildBase():** Amennyiben van elegendő nyersanyag, létrehoz egy bázist az aszteroidán és a felhasznált nyersanyagokat kiveszi az Inventory-kból.
* **+dropMaterial(m:Material):** Amennyiben lehetséges (üres aszteroida átfúrt felszínnel), nyersanyagot helyez az aszteroidába az invetory-ból.
* **+placeStarGate():** Amennyiben van az Inventory-ben StarGate, lehelyezi azt az adott aszteroidára.

Inventory

#### Felelősség

Ez az osztály felel a nyersanyagok, kapuk tárolásáért.

#### Ősosztályok

Printable->Inventory

#### Interfészek

-

#### Attribútumok

* **-id: int:** Ez az adott Inventory id-je, ezt arra használjuk fel, hogy az Inventory-t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* **-material:Material[0…10]** egy tömb ami Material típusu objektumokat táról. Ebben van számontartva, hogy milyen nyersanyagok vannak a jelenlegi Inventory-ban.
* **-starGate:StarGate[0…3]:** egy tömb ami az elkészített StarGate-t tárólja

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Inventory fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+removeMaterial(m:Material):Material** a megadott típusú Material-t kiveszi az Inventory-ból és vissza adja vissza térési értékként.
* **+removeStarGate(): StartGate** kivesz egy StarGate-t az inventory-ból és azt adja vissza mint visszatérési érték.
* **+subSet(i: Invetory): Inventory** ezt a függvényét hívjuk a recpetnek, és azt kapjuk meg, hogy mi hiányzik még a recept elkészítéséhez.
* **+subtraction(i: Inventory):Inventory** ezt a függvényt hívjuk a recepten amikor már ki is akarjuk szedni az adot inventory-ból a releváns nyersanyagokat.
* **addStarGAte(sg:StarGate)**: hozzá ad egy StarGate-t az inventory-hoz.
* **addMaterial(m:Material)**: hozzá adja a megadott Material-t az Inventory-hoz.

Material

#### Felelősség

Ez az absztrakt osztály felel a nyersanyagokért és viselkedésükért.

#### Ősosztályok

Printable->Material

#### Interfészek

-

#### Attribútumok

* **-id: int:** Ez az adott Material id-je, ezt arra használjuk fel, hogy az Material -t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* **#isRadio: boolean:** Ez a változó mondja meg, hogy radioaktív-e az adott nyersanyag.
* **#Name:String:** Ez tárolja az adott nyersanyag nevét.
* **#canEvaporate:boolean:** Ez a változó mondja meg, hogy elpárolog-e a nyersanyag ha napközelbe kerül
* **-exposedCounter:int:** Ez tárolja számon, hogy milyen hosszan volt kitéve a napnak az adott nyersanyag.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Material fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+exposedAndCloseToSun(a:Asteroid):** Ez a függvény hívóik, amikor napközelben felszínen van a nyersanyag. A paraméterül kapott Asteroid aminek a felszínén van. Növeli exposedCounter-t

Iron

#### Felelősség

Ez az osztály felel a vas viselkedésért.

#### Ősosztályok

Printable->Material

#### Interfészek

-

#### Attribútumok

* **-id: int:** Ez az adott Material id-je, ezt arra használjuk fel, hogy az Material -t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* **#isRadio: boolean:** Ez a változó mondja meg, hogy radioaktív-e az adott nyersanyag.
* **#Name:String:** Ez tárolja az adott nyersanyag nevét.
* **#canEvaporate:boolean:** Ez a változó mondja meg, hogy elpárolog-e a nyersanyag ha napközelbe kerül
* **-exposedCounter:int:** Ez tárolja számon, hogy milyen hosszan volt kitéve a napnak az adott nyersanyag.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Material fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+exposedAndCloseToSun(a:Asteroid):** Ez a függvény hívodik, amikor napközelben felszínen van a nyersanyag. A paraméterül kapott Asteroid aminek a felszínén van. Növeli exposedCounter-t

Uran

#### Felelősség

Ez az osztály az Uran viselkedésér-t felel

#### Ősosztályok

Printable->Material

#### Interfészek

-

#### Attribútumok

* **-id: int:** Ez az adott Material id-je, ezt arra használjuk fel, hogy az Material -t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* **#isRadio: boolean:** Ez a változó mondja meg, hogy radioaktív-e az adott nyersanyag.
* **#Name:String:** Ez tárolja az adott nyersanyag nevét.
* **#canEvaporate:boolean:** Ez a változó mondja meg, hogy elpárolog-e a nyersanyag ha napközelbe kerül.
* **-exposedCounter:int:** Ez tárolja számon, hogy milyen hosszan volt kitéve a napnak az adott nyersanyag.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Material fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+exposedAndCloseToSun(a:Asteroid):** Ez a függvény hívodik, amikor napközelben felszínen van a nyersanyag. A paraméterül kapott Asteroid aminek a felszínén van. Növeli exposedCounter-t, ha 3 fölé emelkedik, akkor az asteroid amit paraméterül kapott explode függvényét meghívja.

Ice

#### Felelősség

* Ez az osztály az Ice viselkedésér-t felel

#### Ősosztályok

Printable->Material

#### Interfészek

-

#### Attribútumok

* **-id: int:** Ez az adott Material id-je, ezt arra használjuk fel, hogy az Material -t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* **#isRadio: boolean:** Ez a változó mondja meg, hogy radioaktív-e az adott nyersanyag.
* **#Name:String:** Ez tárolja az adott nyersanyag nevét.
* **#canEvaporate:boolean:** Ez a változó mondja meg, hogy elpárolog-e a nyersanyag ha napközelbe kerül
* **-exposedCounter:int:** Ez tárolja számon, hogy milyen hosszan volt kitéve a napnak az adott nyersanyag.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Material fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+exposedAndCloseToSun(a:Asteroid):** Ez a függvény hívodik, amikor napközelben felszínen van a nyersanyag. A paraméterül kapott Asteroid aminek a felszínén van. Meghívja a paraméterül kapott aszteroida evaporateMaterial függvényét

Coal

#### Felelősség

Ez az osztály felel a coal viselkedésért.

#### Ősosztályok

Printable->Material

#### Interfészek

-

#### Attribútumok

* **-id: int:** Ez az adott Material id-je, ezt arra használjuk fel, hogy az Material -t azonosítani tudjuk.
* **-nextId: int:** Ez egy statikus tagváltozó mely azt tartja számon, hogy mi legyen a következő létrehozott id-t igénylő objektum id-je. Azért statikus, hogy az id-k egyediségét garantáljuk.
* **#isRadio: boolean:** Ez a változó mondja meg, hogy radioaktív-e az adott nyersanyag.
* **#Name:String:** Ez tárolja az adott nyersanyag nevét.
* **#canEvaporate:boolean:** Ez a változó mondja meg, hogy elpárolog-e a nyersanyag ha napközelbe kerül
* **-exposedCounter:int:** Ez tárolja számon, hogy milyen hosszan volt kitéve a napnak az adott nyersanyag.

#### Metódusok

* **+getUIString(): String:** Ez a függvény generálja az adott Material fontos tulajdonságait tartalmazó stringet (ennek legelső eleme az épület id-je) amit a UI megkap s a játékosnak kiír megfelelő parancs esetén.
* **+exposedAndCloseToSun(a:Asteroid):** Ez a függvény hívodik, amikor napközelben felszínen van a nyersanyag. A paraméterül kapott Asteroid aminek a felszínén van. Növeli exposedCounter-t