

Taleen's Inventory Manager

2017. 06.26.

Egy kis bevezető:

Mire jó a szkript?

Gyakorlatilag teljes mértékben automatizálni lehet vele az ércfeldolgozást, öntvénykészítést, valamint az egyes cikkek tárolását, és irányítást is automatizálni lehet. A beállítások elvégzése után az egyetlen további dolog, amire figyelniünk kell, hogy majd potoljuk az ércet, ha akarjuk.

Be lehet állítani minden egyes alkatrésze egy kvótát, egy célértéket, amin a szkript majd igyekszik tartani a darabszámot, és ha fogy belőle valamennyre, pótlódik a készlet. Tehát pl. 5000 Steel plate, 2000 Interior plate, 2000 construction component... Szándékosan igyekszem az egyes neveket megtartani a cikkeknek (alkatrészek, öntvények, érc), mivel a szkript nyelve is angol. Azonban az igyekszem segítséget adni ahhoz, hogy teljesen meg lehessen érteni magyarul is.

Hogyan kell használni?

A szokásos szkript „telepítés” után annyit a dolgunk, hogy a megfelelő blokkok (konténerek, ércfeldolgozó, összeszerelő, stb...) neveibe beelírjunk teg-eket. Ezek alapján a teg-ek alapján tudja majd a szkript, hogy mit, hogyan, milyen csinaljon.

„Telepítés”:

A megszokott mód: egy programozható blokkba betenni a szkriptet, majd azt egy időzítő blokkal folyamatosan futtatni. Azt javaslom, hogy 3-10 mp közötti időtartamként indítsa el, majd önmagát az időzítő blokk.

Angol-magyar:

Az angol név alatt azt tüntetem fel, ahogyan a szkriptben szerepel!

Ércék és öntvények:

Angol név	Magyar név	Angol név	Magyar név
Ore	Érc	Ingot	Öntvény
Cobalt (Co)	Cobalt érc	Cobalt (Co)	Cobalttrúd
Gold (Au)	Arany érc	Gold (Au)	Aranyrúd
Iron (Fe)	Vas érc	Iron (Fe)	Vasrúd
Magnesium (Mg)	Magnézium érc	Magnesium (Mg)	Magnézium por
Nickel (Ni)	Nikkel érc	Nickel (Ni)	Nikkel rúd
Platinum (Pt)	Platina érc	Platinum (Pt)	Platinarúd
Scrap	Fémhulladék		
Silicon (Si)	Szilikon érc	Silicon (Si)	Szilícium Lapka
Silver (Ag)	Ezüst érc	Silver (Ag)	Ezüstrúd
Stone	Kő	Stone	Kavics
Uranium (U)	Úránium érc	Uranium (U)	Úránium rúd

Amit itt fontos észrevenni, hogy angolul az ércék és öntvények nevei ugyan azok! (Később meglátod miért fontos ez.)

Angol név	Magyar név	Angol név	Magyar név
Component	Alkatrészek	AmmoMagazine	Lőszer tárák
BulletproofGlass	Golyóálló üveg	Missile200mm	200mm rakétatároló
Computer	Számítógép	NATO_25x184mm	25x184mm NATO
Construction	Építő elem	NATO_5p56x45mm	5.56x45mm NATO tár
Detector	Észlelő Elem	Egyéb	Egyéb
Display	Kijelző	HydrogenBottle	Hidrogén palack
Explosives	Robbanóanyagok	OxygenBottle	Oxigén palack
Girder	Gerenda	Welder	Hegesztő
GravityGen	Gravitációs Generátor	Grinder	Bontó
InteriorPlate	Beltéri Lemez	Hand drill	Kézi fúró
LargeTube	Nagy Acél Cső	Rifle	Géppuska
Medical	Orvosi	QUOTA	kvóta (célérték)
Motor	Felszerelések		
MetalGrid	Fém rács	AUTO	automatikus
PowerCell	Motorja	STATUS	státusz (állapot)
RadioComm	Rádió-kommunikációs Felszerelések	teg	tag (jelzés)
Reactor	Reaktor elemek		
SmallTube	Kis acél cső		
SolarCell	Napelem		
SteelPlate	Acél Lemez		
Superconductor	Superconductor Conduits (Szupervezető vezeték)		
Thrust	Hajtómű alkatrészek		

Ezeket nem kell megtanulni!! A kvóta panelekkel (később), bármikor előhozhatók a játékomban belül is.

Használat:

A következőkben igyekszem funkciókat szerint leírni mire lehet használni a szkriptet.

Alapjáról a teg-ek így néznek ki: [TIM ...], ahol a pontok az átalunk használni kívánt funkciókat jelölnék.

Tehát pl van egy „Nagy tárolókonténer” blokkod. A névét ír át erre: „Nagy tárolókonténer [TIM Ingot]” vagy erre: „[TIM Ingot] Nagy tárolókonténer”, azaz tőkmindegy a teg hol van a névében.

1. Tárolás konténerekben:

[TIM Ingot] Abban a konténerben, aminek a nevében ez szerepel, csak öntvények fognak tárolódni!

[TIM Ore] Abban a konténerben, aminek a nevében ez szerepel, csak ércék fognak tárolódni!

[TIM Ore Ingot] Abban a konténerben, aminek a nevében ez szerepel, ércék és öntvények fognak tárolódni! **Szokásból vannak elválasztva az egyes cikkek.**

[TIM Ore Ingot Component] Abban a konténerben, aminek a nevében ez szerepel, ércék, öntvények és alkatrészek fognak tárolódni!

[TIM Ore Ingot SteelPlate] Abban a konténerben, aminek a nevében ez szerepel, tárolódní fog az összes érc, az összes öntvény, az alkatrészek közül pedig csak a SteelPlate (acél lemez).

[TIM Iron SteelPlate] Tárolódní fog benne egyaránt a vasérc, valamint a vasöntvény is! (hiszen angolul ugyan az a kettő), és a SteelPlate.

[TIM Ingot/Iron SteelPlate] Csak a vas öntvény (öntvény/vas) és a SteelPlate tárolódní majd benne!

[TIM Ingot/Uranium] Ha reaktor nevére írod ezt be, akkor abban a reaktorban fog tárolódní az uránium! Vigyázz! Ha konténer nevébe teszed ezt be, akkor előfordul, hogy az összes reaktorból abban a konténerbe megy az uránium, és a bázis egyszer csak lekapcsol. (már ha nincs egyetlen akksi sem rajta ugyan)

2. Tárolás konténerekben prioritással:

Nem csak azt lehet megadni, hogy mi hol tárolódní, hanem azt is, hogy mennyi tárolódní, hanem azt is, hogy milyen fontos az, hogy abból a valamiből adott mennyiség abban a tárolóban legyen.

Prioritás jelzője a P1, P2, P3... egy kettőspont után, ahol a P utáni szám a fontosságát jelöli. Elsődleges prioritás, másodlagos, harmadlagos... Vagyis minél kisebb, annál inkább fontosabb, hogy abban a konténerben legyen a cikk. Maximum 1000-ig kezel a prioritást a szkript, szóval lehet ügyeskedni :D

A, Van a bázison 2 konténer:

Az egyik nevében a [TIM Ore Ingot Component] szerepel, míg a másik nevében a [TIM SteelPlate:P1] teg. Ebben az esetben minden az 1-es konténerben fog tárolódní (az egy általános tároló), kivéve a SteelPlate-et, ami megy a második tárolóba, hiszen annak nagyobb a prioritása. Igazság szerint, ha nincs az 1-es nevében prioritás tag, akkor az mindig a legkisebb prioritású lesz.

B, Tegyük fel, van egy **bányászahajó!** A drillek (fúrók) nevében nincs teg, míg a bányászahajó konténerében szerepel a [TIM Ore:P900] teg. Ez azt jelenti, hogy a fúrák után mennék az ércék a konténerbe. Visszamész a bázisodra, és bedokkolok. A bázis egy konténerben szerepel az a teg, hogy [TIM Ore:P100 Ingot:P100 Component:P100]. A 100 fontosabb prioritás a 900-nál, tehát az érc át fog menni a bányászahajóról a bázisod konténerébe, a bedokkolásod után pár másodperccel, automatikusan! Aztán csak kidokkolok, és mehetsz is vissza bányászni.

3. Tárolás konténerekben prioritással és darabszámmal:

A darabszám a prioritással karöltve csodákra képes! :D A darabszám jelzője egy szám, egy kettőspont után.

A, Tegyük fel, van egy **drónod, bányászahajó, stb. Oda kell figyelned arra, hogy az uránium ne fogyni ki a reaktorból.** Kivéve, ha van egy [TIM Ore Ingot] blokkod, ami az urániumot tárolja, és egy ilyen teg: [TIM Ingot/Uranium:P1.5], míg a bázisod egy konténerében ez [TIM Ingot/Uranium:P2] és a bázisod egy reaktorában ez: [TIM Ingot/Uranium:P1:100].

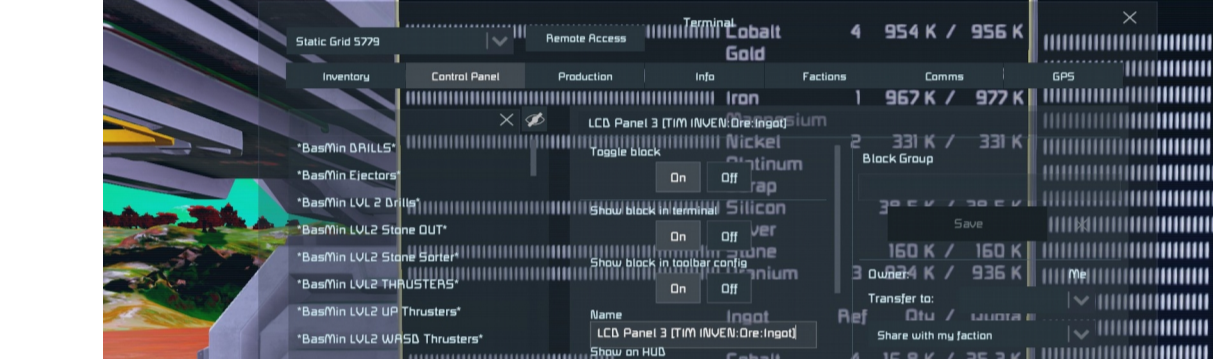
Gratulálók! Ezzel meg is van az **energia menedzselő rendszered!** Mit történik ugyanis? Ami összes uránium taláha is finomításra kerül, abból 100 mindig a nagy hajó reaktorában van, a kis hajó reaktorai mindig újratöltik, egyenként 5-5 urániummal, és a maradék meg tárolódik a nagy hajó konténerében. 5-5 uránium a kis reaktorokban elég kb. 1 órárs repülésre, és amint visszamész a bázisra, pár másodpercen belül újra is töltődik!

B, Ismét egy kis **bányászahajós** téma! A darabszámokkal már megoldható egy **sorternek elküldi kökődobó rendszer** is. Sőt! Többé olyan probléma sem fordulhat elő, hogy a bányászahajó a bázisodra dokkolva az elküldi kidobálni a bázisod konténerében levő követ is, mert **előrejeltehető lekapcsolni a sorternek kökődobó rendszer** a bányászahajó a dokkolás előtt. A bányászahajó ejectorában, vagy konnektorjában, amiket a kidobásra annak használni, szerepel az a teg: [TIM Ore/Stone:P800:200]. A bányászahajó konténeré: [TIM Ore:P900], míg a bázisod konténeré: [TIM Ore:P100 Ingot:P100 Component:P100]. Mi fog történni?

A kibányászott ércet közül az értékesek mennek a bányászahajó konténerébe, míg a kő megy a kidobókba, hiszen azoknak magasabb a prioritása. (Ne felejtsek meg beállítani, hogy a kidobók dobják ki a beléjük került ércet! Throw Out ON (kidobás ON, ezert ugyanis nem a TIM fele). Amikor bedokkolok a bázisodra, akkor az ércet mennek a bázisod konténerébe, hiszen annak van a legnagyobb prioritása, viszont a bázisodban levő kő nem megy a kidobókba!

C, Tegyük fel, vagy egy **hegesztő/szétbontó hajó!** Repülés egy kicsit, hegesztés vele, majd miután kifogytak az alkatrészek (vagy tele vagy), mehetsz vissza a bázisra újratölteni/kiüríteni. Aztán megint egy kis repkedés, majd újra alkatrészek pakolgatás... A pakolgatás elég egyszerű és unalmas dolog. DE! Megtudod azt tenni, hogy a **hegesztő hajó konténerében magas prioritással mindig adott számú alkatrészt tárolódní.** Ha kifogyott, akkor feltöltődik, ha tele van a kis hajó, akkor meg kiürül automatikusan!

Az alábbi problémamegoldásra a BEE (méh) nevű atmoszférikus hegesztő hajómat hozom fel.



A központi nagy konténerének ez a teljes neve:

BEE Large Cargo Container [TIM BulletproofGlass:P1400 Computer:P1300 Construction:P1500 Detector:P1:25 Display:P1:40 Explosives:P1:60 Girder:P1:200 GravityGenerator:P1:6 InteriorPlate:P1:500 LargeTube:P1:200 Medical:P1:25 MetalGrid:P1:200 Motor:P1:1500 PowerCell:P1:360 RadioCommunication:P1:40 Reactor:P1:100 SmallTube:P1:500 SolarCell:P1:320 SteelPlate:P1:5000 Superconductor:P1:100 Thrust:P1:50]

1-es prioritást kaptak, tehát mindig újratöltődnek amint a bázisra csatlakozok vele. (A sok hajtóművel pedig piszok gyorsan lehet repülni, így nagyon hasznos kis gépezet.)

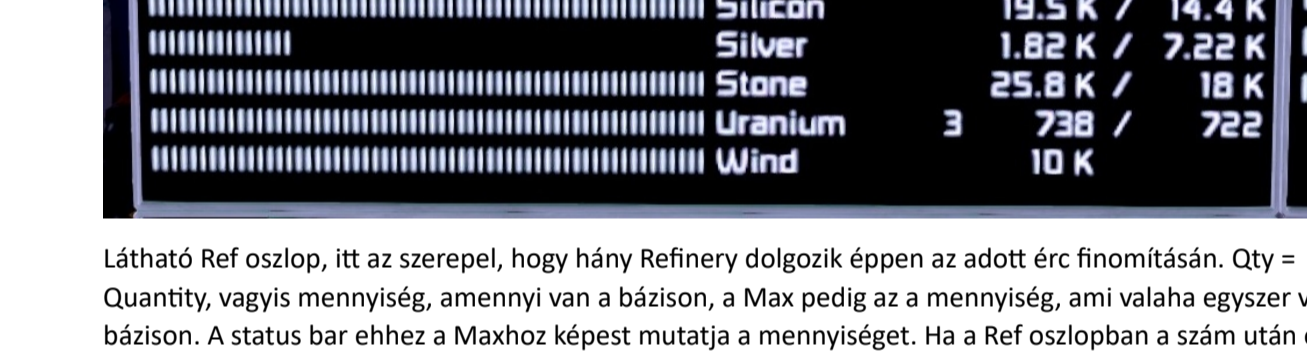
4. LCD használat:

Létre lehet hozni információs paneleket (INVEN, STATUS és DEBUG parancs), valamint a kvóta panel is LCD-n megy. Az információs panelek segítségével az aktuális cikk mennyiségét tudod megjelteni, ami a bázison van (INVEN), vagy a szkripttel kapcsolatos infókat (STATUS, DEBUG), míg a kvóta panel segítségével meg tudod adni a célértéket, aminek a legyártására törekedni fog a TIM.

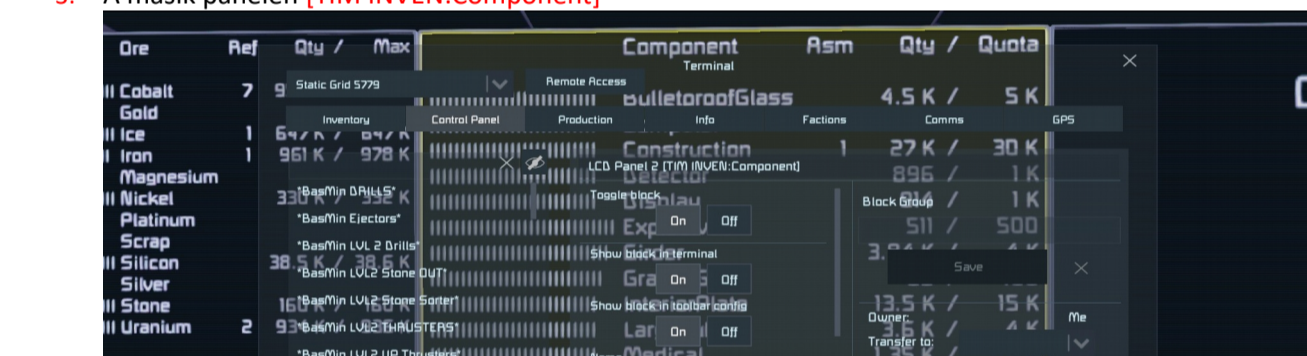
Információs panelek:

1. [TIM INVEN] Ezzel a parancssal túl sok információ kerül az LCD-re, így nem láthatod teljesen. A teljes Inventory mennyiség listázza lenne. Sokkal inkább javasolom helyette a szűkítést, az egyik LCD panelen csak az Ore és Ingot legyen megjelölve, a másikon pedig csak a Component-ek.

2. [TIM INVEN:Ore:Ingot]



És az eredménye ez lesz:



Látható Ref oszlop, itt az szerepel, hogy hány Refinery dolgozik éppen az adott érc finomításán. Qty = Quantity, vagyis mennyiség, amennyi van a bázison, a Max pedig az a mennyiség, ami valaha egyszer volt a bázison. A status bar ehhez a Maxhoz képest mutatja a mennyiséget. Ha a Ref oszlopban a szám után egy felkiáltójel látható, az azt jelenti, hogy az adott érc finomítódki elsődleges prioritással, mert abból nagy hiány alakult ki hirtelen. (Persze csak akkor, ha van hozzá érc is.)

3. A másik panelet [TIM INVEN:Component]



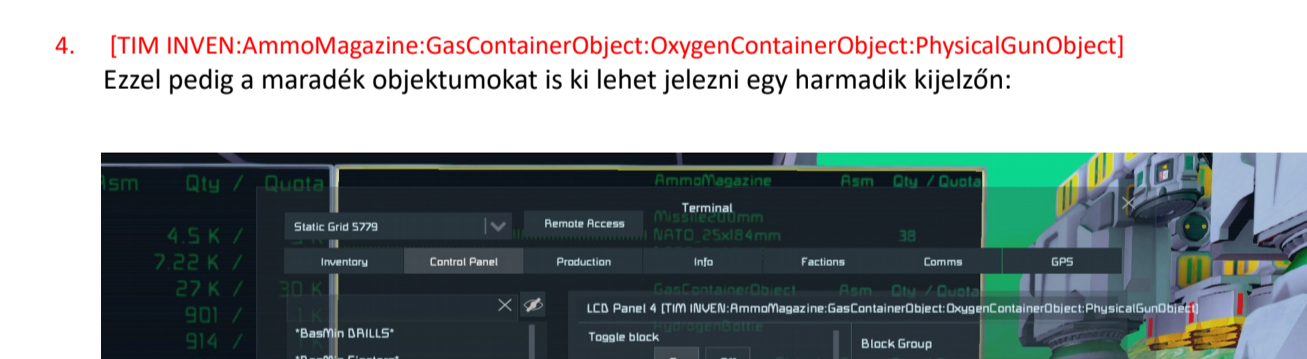
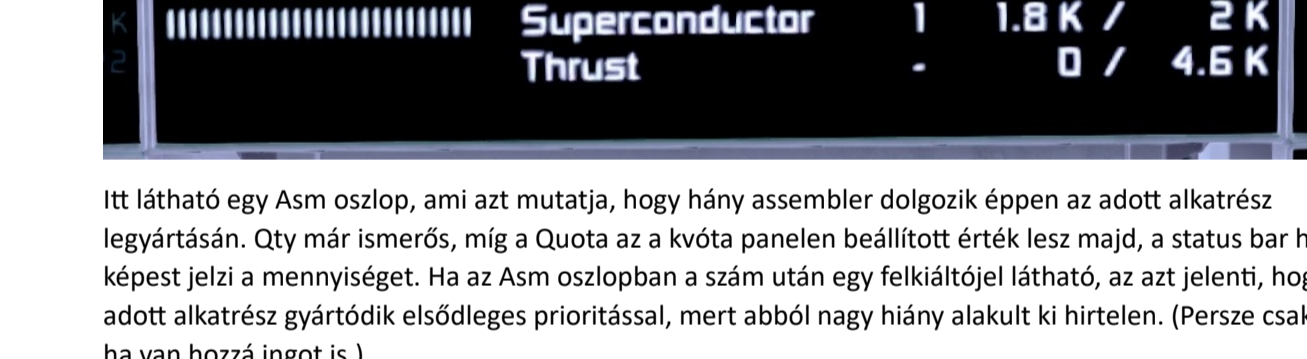
És az eredménye:



Itt látható egy Asm oszlop, ami azt mutatja, hogy hány assembler dolgozik éppen az adott alkatrészt legyártásán. Qty már ismerős, míg a Quota az a kvóta panelen beállított érték lesz majd, a status bar hozzá képest jelzi a mennyiséget. Ha az Asm oszlopban a szám után egy felkiáltójel látható, az azt jelenti, hogy az adott alkatrészt gyártódki elsődleges prioritással, mert abból nagy hiány alakult ki hirtelen. (Persze csak akkor, ha van hozzá ingot is.)

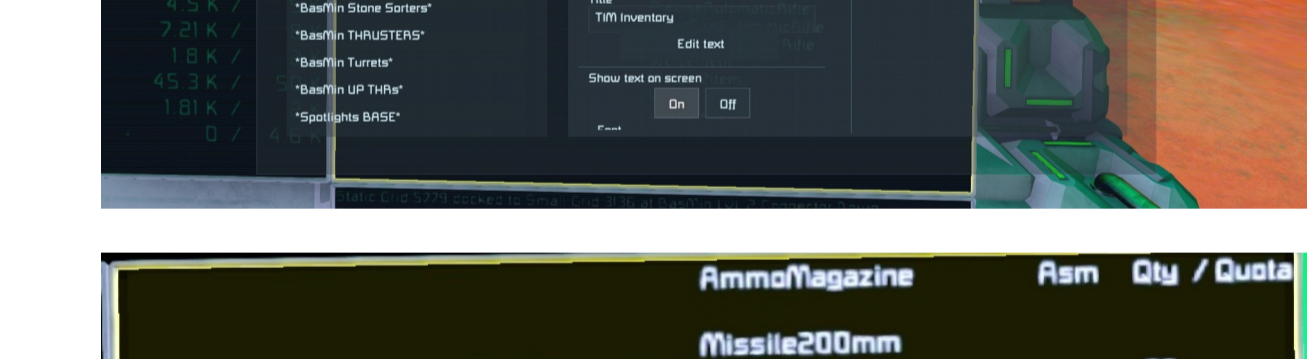
4. [TIM INVEN:AmmoMagazine:GasContainerObject:OxygenContainerObject:PhysicalGunObject]

Ezzel pedig a maradék objektumokat is ki lehet jelezni egy harmadik kijelzőn:



Nem lehet szűkíteni tovább is a kört, hogy csak az egyes cikkek mennyiségét jelenjen meg, bár bevallom nekem még nem is volt szükségem ennek a használatára. Ha mégis szükséges lenne az, pl. egy „nyomtatható” rakéta megmaradt alkatrészeinek kijelzése...), akkor az inkább az Automatic LCDs 2 szkripttel oldtam meg, mert az az egyes konténerekre is lehet szűrni. A TIM az inkább hajó/bázis/conténerének tartalomát összegezve jeleníti csak meg, és az INVEN parancs után is csak a fentebb leírt dolgokat lehet írni.

5. [TIM STATUS]



Run: hanyadjára futott le a szkript mióta elindítottad.

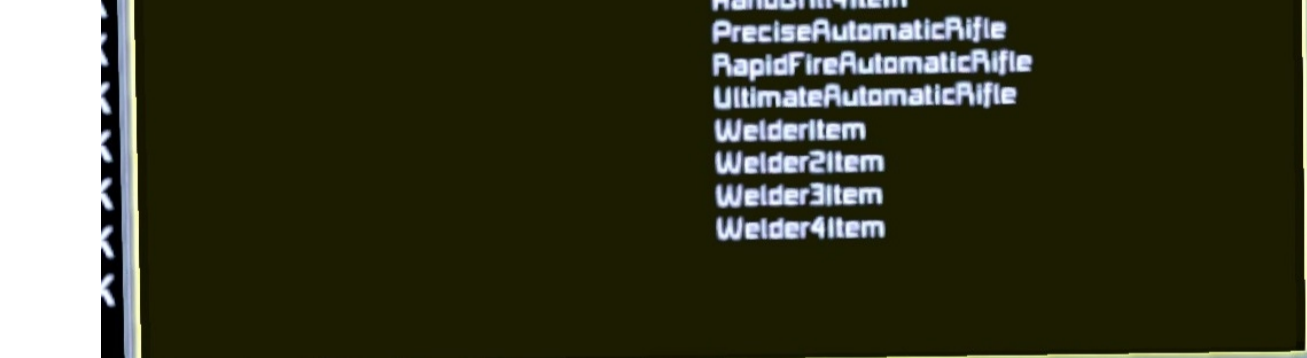
Stop: a feladatokat hány lépésben ha újra végre hajtja a feladat, akkor több lépés kell a végrehajtáshoz, ilyenkor 1 lefutás 1 lépést jelent, és ha pl. 5 lépés állítasz be, akkor 5 lefutás alatt fogja végrehajtani a feladatotsort.)

Time: a lefutáshoz szükséges idő mennyi volt

Load: Hány %-os a szkript terheltsége. 100%-hoz közel a szkript már gyakran leáll! „A szkript túl komplex” parancssal, túl sok feladatot kell végre hajtani. Ilyenkor manuálisan kell újraindítani, beletelni a programozható blokkba, ellenőrzés, majd kijelzés. Erdemes a terheltséget kb. 75% alatt tartani, ezt úgy lehet megtenni, hogy több lépést hozol létre, így az egyes lépésekben kevesebb feladatot kell végrehajtania a TIM-nek. Több lépést létrehozni 2 fázisban lehet, vagy úgy egy belépős a célterület és a „cycle=1” srtt áford 1-nél nagyobb számr, vagy pedig egy „cycle=#” argumentet futtatsz a szkriptben. (Ahol # a szám, amennyi lépést szeretnél.)

S, R, A: nem tudom mit jelenthetnek, még a szkript leírásában sem említi Taleen... Úgy sejttem ez az egyes feladatokat jelentené. S=Sorting, vagyis hogy hány cikket mozgattam meg, vagy ellenőrzött le, hogy a helyén van-e, R=Refinery, vagyis hány parancsot adott ki a finomítóknak, A=Assembler, vagyis hogy hány parancsot adott ki az összeszerelőnek.

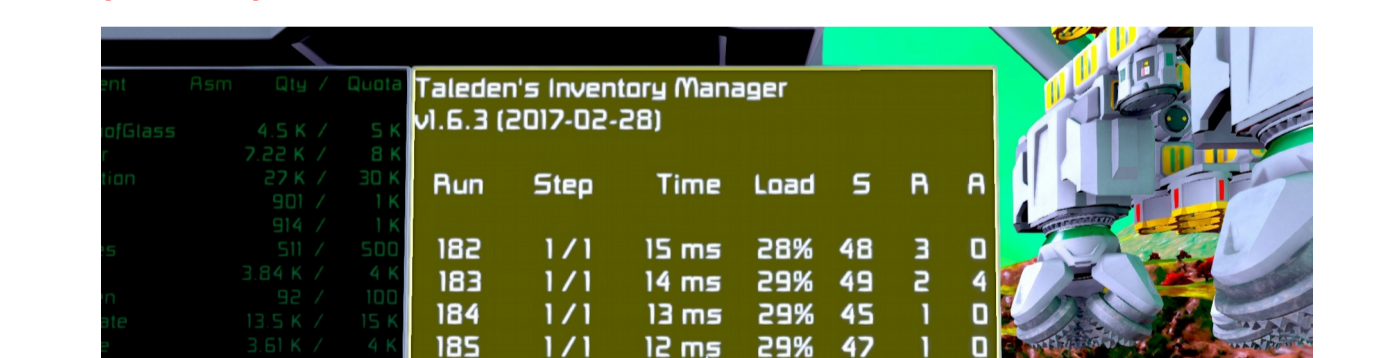
6. [TIM DEBUG]



Nagyon jó kis dolgokra hívhatjuk fel a figyelmet, pl. ha nem lehet eljutni egyik konténerből a másókba, milyen dolgok dőlhetnek be, és kijelző az egy-egy elem helyére. Ehhez már tényleg kell egy kis angol.

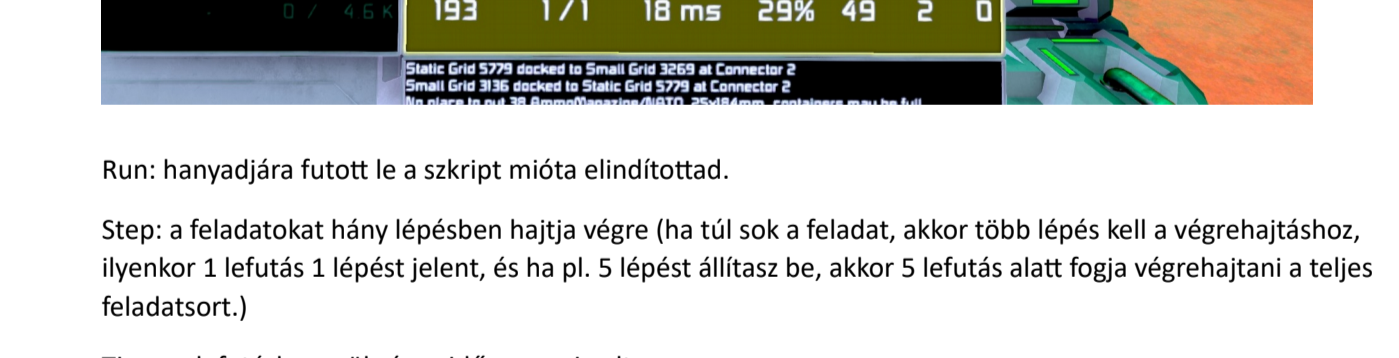
Quóta panel:

1. [TIM QUOTA]



A kvóta panelek lehet módosítani ugye a célértékeket. A szkript az assembler-ek segítségével igyekszik ezen a célértéken tartani a legyártott alkatrészek mennyiségét. Vagy legalábbis a 95-105%-k között. Minél kevesebb van egy adott alkatrészt %-os mennyiségéből a kvótához képest, annál nagyobb prioritással fog majd az az assembler gyártóórára kerülni. Ahány assembler dolgozik az adott alkatrészen, az meg főleg majd megy az INVEN:Component panelek is. Az ércnek azért nincs kvótája, mert abból mindig az összes lefinomításra kerül, így a bázison a bázison valaha volt Max. érték jelenik meg. Az alkatrészek gyártásánál is meg lehet szűrni a kvótát, ha futtatjuk a „quota=literal” argumentet a szkriptben, vagy beállítjuk a szkriptben módosított a megfelelő sort.) Illyenkor arra lehet fogyni majd a szkript, hogy minden ingotból alkatrészt gyártson. Ha vissza akarod csinálni, akkor a „quota=stable” argumentet kell futtatni.

A kvótákat így kell módosítani, hogy belépés a szövegdobozába az LCD panelekn, és ott módosítgató a 2. oszlopban szereplő számokat. (Tök mindegy mennyi szökököt haegysz majd előtte-utána, a szkript helyrehozza majd.)



A 3. oszlopban, a %-os értékeknek a „quota=literal” módban van értelme. Hogy is működik ez a literal mód? Amint a alkatrészt elérni a beállított mennyiséget, elkezd ugye egy másik alkatrészt gyártóódní. Egy idő után azonban minden alkatrészből meglesz az előre beállított mennyiség. Ekkor a szkript nem fog tovább ülni tényleen, ha látja hogy van ingot, akkor megneveli az adott alkatrészt kvótáját a 3. oszlopba írt %-os értékkel. Majd vezérele az assembler-eket az új célérték előállítására **torozsik. Ha elérte, egy kövözet alkatrészeknek növeli meg a kvótáját, és így szépen elfogynek majd az ingot-ok.** A „quota=stable” módban pedig az a %-os növegtés megváltozik, szóval ott a kvóták stabilak. Jőmagam a stabil módot szoktam használni, és a kvótákat mindig kézzel lövöm be attól függően, hogy hol tartok a fejlődésben.

A Quóta panel szövegdobozában egyébként lehet módosítani az ingot-ok célértékeit is, azonban az csak addig lesz érvényes, amíg meg nem haladja azt a feladott mennyiséget. Ezután ismét a Max mennyiség lesz az egy kvóta is, és a progress bar is ahhoz fog igazodni. Persze van azért 2 haszna is a dolgoknak: Ha megnevelöd pár dolgonak a kvótáját, akkor egy ideig csak azok fognak feloldogódní, így ha tudod hogy pl. kobaltból jó sok kell majd nemcsak, akkora előre megnevelheted a mennyiségét a többithez képest. (Egy Nanite Control Factory gyártása előtt pl.). Egy másik előnye, hogy az uránium hasonlóan viselkedik az alkatrészekhez, így a nála megadott kvótát a TIM nagy prioritással fogja figyelni, és ha fogy a mennyisége, akkor gyorsan vissza pótolja.

5. Refinery (ércfinomító) menedzselés

A menedzselés nem csak a Refinery-re, hanem az Arc Furnace (ívkemence) is kiterjed, valamint a módolt finomítókra is, pl. Naquadan feloldozó (stagan mod). [TIM AUTO] a teg, amivel a TIM szkript irányítja alá kerül a Refinery. A TIM megnezi melyik ingotból is lehet a legkevesebb (a korábbi maximumaikhoz képest) és azt fogja feloldoztatni a Refinery-ben. Vagy egy kivétel, az uránium, amire meg lehet adni kvótát is, hogy sose fogyhass ki belőle!

Működési elve: a TIM pár cikksal alatt kitapasztálja, hogy mekkora mennyiségű ércet kell betennie a Refinery-be, hogy meg legyen bennük újra a követő ciklus lefut. (Ha több lépés adsz meg neki a „cycle=#” argumenttel, ahhoz is alkalmazkodik fog, pl. 5 lépésnél el érjedt mennyiség 5-szöröse lesz bele a finomítóba, hiszem majd csak 5-ször annyit idő múlva is fog visszanézni rá) Azt jelenti, hogy a Refinery-kben egyszerre kevés ér lesz, pár 100 pár ezer db, viszont ez egyben azt is jelenti, hogy rákukkantva mondjuk az érc-tároló konténerre, pontosan láthatod, mennyi érced van még. Valamint ha az ércet akkor gyorsan le is foggesztöd a hatásra is, a Refinery-kből hamar kifogy a bennük levő mennyiség és pakolthatódní nyugodtan manuálisan amint szeretnél.

Meg lehet adni azt is, hogy az egyes finomítódki elsődleges mit dolgozzanak fel! Vagyis ha van adott érc a bázison, akkor az a finomító csak azt az ércet fogja feloldozni, ha megadod a teg-ét, az egy elsődlegesben Platínát és Úrániumot „feldolgozó akardó” finomító lesz. (Bármilyen más ércet is lehetne természetesen, és nem is csak kettőt.)

[TIM AUTO:Platinum:Uranium] lesz például egy ilyen módok a teg-je, az egy elsődlegesben Platínát és Úrániumot „feldolgozó akardó” finomító lesz. (Bármilyen más ércet is lehetne természetesen, és nem is csak kettőt.)

Magában a Refinery név teg-ében is megadható egy célértéket. Ez a célérték ráadásul fontosabb lesz, mint a kvóta panelek megadott. Ez azt jelenti, hogy a finomító addig dolgozik az adott ércen, amíg el nem éri az ingot-ja a [TIM AUTO:mennyiség]. Ha ez megvan, akkor utána visszatér az AUTO parancshoz.

[TIM Uranium:100] teg-el gondoskodhatsz arról, hogy autói garátnál legyen 100 urániumad a bázison, mindig! Figyelj rá, hogy az AUTO itt a végén van, nem az elején!

6. Assembler (összeszerelő) menedzselés:

Itt is a **[TIM AUTO]** lesz a teg, ebben hasonlít a Refinery-kre. A QUOTA panelen megadott célértékeket figyeli csak! Itt nem adható meg az egyes Assembler-ekre elsődlegesen előállítandó alkatrész, felesleges is, hiszen a kvótától való csökkenés esetén rááll az összes Assembler a gyártásra. Minél kevesebb az adott alkatrész mennyisége a QUOTA panelen megadott %-os célértékhez képest, annál nagyobb prioritással fog legyártódni az alkatrész. A TIM okos annyira, hogy ha nincs pl. platínád, akkor nem gyárt Thruster Component-et. (Persze ez nem teljesen igaz, néha bepróbálkozik azért, de ha nem kezd el gyártódni az Assemblerben, akkor 2-3 ciklus után ki is veszi a gyártásorból!) Ugyan ez igaz minden más alkatrésze is, tehát pl. ha nincs szilikonod, akkor nem fog gyártódni SolarCell, vagy Computer.

A beállított célérték 95-105%-a között igyekszik tartani az alkatrészek mennyiségét, azonban érdemes megemlíteni a „**quota=literal**” és „**quota=stable**” argumenteket, amikről a QUOTA panelhez visszalapozva olvashatsz (alig 1 oldalal fentebb!)

7. Egyéb hasznos info:

Redundancia, azaz hibatűrés: A legtöbb cikk nevére a TIM tartalmaz szinonimákat is. Tehát pl. ha beírod, hogy 200mm, vagy missile akkor az kijavítódik Missile200mm-re. Vagyis pl. ez: **[TIM 200mm:5]** egy rakétavető ágyú nevében, kijavítódik erre: **[TIM Missile200mm:5]**, így biztosítva lesz az ágyúd folyamatos újratöltése a konténerekből! (Plusz ha leszedik az ágyúdat, nem is vész oda sok rakétád, max. csak 5 db.:D

Ákármenyi TIM-et rakhatsz a hajódra, futhat egy a kis hajódon is, a bázisonod is, a nagy hajódon is. Ha csatlakoznak, akkor csak az egyik marad aktív, a második (harmadik, negyedik, ötödik...) blokk nem fog lefutni, mert érzékeli hogy már van egy aktív TIM rajta kívül. Ezzel kímélheted a játékot is.

Összekapcsolódott hajók: alapesetben a TIM ugye a kijelzőkre minden konténer minden tartalmát összegezve ír ki. Azonban mi van ha csak azt szeretném, hogy csak a bázis tartalmát jelenítse meg? Ebben segít a **DOCK** parancs. A kapcsolódó konnektor nevébe kell írni egy teg-et, amiben a DOCK van. A DOCK után kettősponttal a másik hajó neve jön, ami arra a konnektorra kapcsolódott rá. Ha a hajó nevében nem szerepel a DOCK után megadott szó, akkor a hajó konténerének tartalma nem adódik hozzá a bázis konténerének tartalmához a kijelzőkön! Amúgy a DEBUG képernyőn látszani is szokott az első sorban a csatlakozott hajókról infó.

Ha van egy **BANYASZHAJO** nevű hajód, akkor annak csak akkor fog beleszámítani a konténer tartalma, ha az alábbi nevű konnektoron keresztül kapcsolódik a bázisodhoz: **[TIM DOCK:BANYASZHAJO]**. Vagy ha ugye nincs a konnektor nevében semmi teg! :D Tehát még egyszer, ha a bázison ha a bázisonod fut az elsődleges TIM, akkor a bázisonod levő konnektor nevében legyen benne a teg, nem a bányászhajó nevében.

Persze ugyan ezt fordítva is el lehet játszani. Ha a bányászhajódon fut az elsődleges TIM, akkor a bányászhajó konnektorának nevébe írd be a DOCK parancsot teg-et. És ha ebben a teg-ben nem fog szerepelni a bázisod neve, akkor a bányászhajó kijelzőin csakis a bányászhajó konténerében levő cikkek fognak kilistázódni.

Mindentől függetlenül, **mivel ugye csak egy TIM fog futni így mind a két hajóra,** a TIM a feladatait továbbra is ellátja, mozgatja az anyagokat, stb..., csak a kijelző adataiba nem fog beleszámítódni a másik hajón levő mennyiség. Plusz a kvótákba sem számolódik bele, szóval átmehet egy **összeszerelő hajóra simán 5000 Steel Plate**, ha beállítod, hogy az ne számoldjon bele a bázisod kijelzőin szereplő adataiba, akkor az összeszerelő hajó feltöltése után az Assemblerek gyorsan igyekeznek visszapótolni majd azt az 5000-es hiányt, még azelőtt mielőtt lekapcsolódná a bázisról. Ha nem bajlódás a DOCK parancsja), akkor majd csak azután gyártódik vissza az 5000-es hiány, miután lekapcsolódtál a bázisról az összeszerelő hajóddal.

8. Szkript Argumentek:

1. cycle=#

Hány lépésből álljon egy ciklus. A kettőskereszt ugye egy számot jelöl. A ciklus felosztása ugyebár a terhelés csökkentése miatt fontos, mert 100%-os terheltség fölött a szkript néha összeomlik. Érdemes 75% alatt tartani. (lásd. STATUS LCD)

2. rewrite

Valami tag átírást engedélyezni. Bár bevallom nem tudtam rájönni mire kellene az a beállítás.

3. norewrite

Inaktívalhatod vele a tag átírás lehetőségét, de ezen kívül nem tudok többet mondani róla.

4. tags=AB

Ha meg akarod változtatni a kapcsos zárójeleket más kezdő és záró karakterekre. Ha a fentit beírod, akkor nem [-vel fog kezdődni és]-re végződni egy tag, hanem A-val kezdődik majd és B-re végződik. **ATIM Ore IngotB**. Alapbeállításra visszaállítani ugye: tags=[] argumenttel, így visszatérhatsz ehhez: **[TIM Ore Ingot]**

5. prefix=PPX

A TIM tag átírását engedélyezheted vele majd a szkripten belül. (gy pl. több szkriptet is lehet majd futtatni egyszerre, az egyiket TIM névvel, a másikat PFX névvel, sőtöbbit...) Alapbeállítás: prefix=TIM

6. scan=type

Engedélyezheted vele az egyes blokk típusokra való keresést is: "Collectors" (begyűjtő blokkok), "Drills" (fúrófejek), "Grinders" (flexek) "Welders" (hegesztők); Alapesetben ezeket a blokkokat a TIM nem skenneli, csak a tárolókat, mivel általában nagyon nagy számban vannak jelen, ami a terheltséget növelné. Azonban ha sorter nélkül akarsz mondjuk érceket kiszívni a fúrófejekből, akkor érdemes egy scan=drills argumentet lefuttatni. Alapbeállítás: scan=none

7. quota=mode

A stable és literal módok között váltás. A mode helyébe kell beírni ezt a 2 szót, infó róluk az Assembler és QUOTA LCD szakasznál.

8. debug=type

Hibakereső szubrutinok futtatása, hogy kiderüljön ha valami gond van a szkripttel, különösen egyéb modok használatakor. A type helyére mehet „quotas” „sorting” „refineries” „assemblers” (Még sosem kellett használnom, így a nyers fordításnál többet nem tudok rólu mondani.)

Vége

Készítette: **lac14321**