

Java Challenge harmadik forduló

játékszabályai v1.2

Változások az 1.1 verzió óta

- *egyértelműsítés: egy bolygón csak 1 vagy 2 csomag lehet*
- *prezentációt a többi csapat nem látja*

Változások az 1.0 verzió óta

- *az akna a saját hajót is felrobbantja*

Környezet

A harmadik fordulóban is egy bolygóközi csomagszállító űrhajót kell irányítanotok a játékszervernek küldött parancsok segítségével. A bolygók egy síkban vannak, így egy kétdimenziós térképen kell navigálni. Ebben a fordulóban minden csapat ugyanazon a pályán versenyez, egymást űrhajóit látják, fel tudják venni a másik csapat által lerakott csomagot is. A játék kezdetén a következő feltételek adottak:

- minden bolygón 1 darab gazdátlan csomag van
- minden csapat 3 űrhajót irányíthat
- minden csapatnak 10 telepíthető aknája van
- minden csapat minden űrhajója ugyanazon a véletlenszerűen kiválasztott bolygón tartózkodik
- minden űrhajó raktere üres
- minden csapat pontszáma 0

A játék során bármikor lekérdezhetők a következő adatok:

- bolygók neve és pozíciója (x, y)
- melyik bolygón van csomag, és ki a csomag tulajdonosa
- hol vannak a többi csapat űrhajói
- a saját űrhajók helyzetét, és a szállított csomagokat

A csomagokkal és bolygókkal kapcsolatban a következő elnevezéseket használjuk:

- csomag tulajdonosa: az a csapat, amely az adott csomagot utoljára egy bolygón felvette, majd egy másik bolygón lerakta. Ha egy másik csapat ezt a csomagot felveszi és egy másik bolygón lerakja, ő lesz az új tulajdonos
- bolygó tulajdonosa: az a csapat, akinek a csomagja a bolygón van
- pontszám: minden csapatnak annyi pontja van, ahány csomagnak ő a tulajdonosa, bolygóinak száma ennél nem nagyobb.¹

Az űrhajó a következő alapvető műveletek elvégzésére képes:

- repülés egyik bolygóról a másikra
 - az űrhajó mindig légvonalban repül

¹ Ha egy csapat bolygójáról felvették a csomagját, a bolygó már nem az övé, de a csomag még igen, amíg le nem rakják egy másik bolygón.

- az űrhajó sebessége állandó (nincs gyorsítás/lassítás), de ha csomagot szállít, akkor lassabb
- ha az űrhajó egyszer elindult a célállomás felé, ez a művelet már nem megszakítható, meg kell várni, míg odaér, az űrhajó menet közben sehol sem áll meg, csak a célállomáson
- az űrhajók nem befolyásolják egymás mozgását (nem ütköznek össze)
- csomag felvétele
 - csak akkor lehetséges, ha van még hely a raktérben
 - a felvett csomag eltűnik a bolygóról, átkerül a hajó rakterébe
 - az űrhajó rakterében 1 csomag fér el
 - ha a bolygón akna van, az űrhajó a csomagot nem veszi fel, az akna felrobban
- csomag lerakása
 - ha a bolygón akna van, az űrhajó a csomagot nem rakja le, az akna felrobban
 - csak akkor lehetséges, ha az adott bolygón még nincs bármely csapat által odaszállított csomag (tehát ha a bolygó üres vagy csak egy gazdátlan csomag van ott)²
 - ha a csomagot másik bolygón rakjuk le, mint ahol felvettük, 1 pontot kapunk, ha a lerakott csomag korábban más csapat tulajdonában volt, a másik csapattól levonásra kerül 1 pont
 - ha a csomagot ott rakjuk le, ahol felvettük, nem kapunk pontot
 - lerakás után a csomag az űrhajó rakteréből a bolygóra kerül
- akna telepítése
 - csak akkor lehetséges, ha van még lerakható aknánk
 - csak arra a bolygóra rakhatunk aknát, ami a miénk

Önmagában azzal, hogy egy bolygóhoz elmegyünk, még nem robban fel az ott lévő akna, csak csomag lerakás/felvétel esetén. Ha az akna felrobbant, az adott űrhajó 2 percig javításra szorul és már az aknarobbanást kiváltó parancsot sem teljesíti. A javítás időtartama alatt a hajó semmilyen parancsot nem teljesít. Az általunk lerakott akna a saját hajónkat is felrobbanthatja, ugyanúgy, mint a többiekét.

A kliens által küldött parancsok, amennyiben megfelelnek a feltételeknek, azonnal végrehajtódnak. A kliens korlátlan számú kérést indíthat a szerver felé, de a kérések közt 100 ms időnek el kell telnie.

A játék menete

A teszt játékszerver indulása hétvégére (2014. 11. 29.) várható, erről majd értesítést küldünk. A döntőn mindenki a saját eszközeivel vesz részt, azokon futtatja majd a kliens programját, itt több fordulóra lehet számítani, ezek egyenként 10-15 percesek lesznek. Az utolsó három forduló eredményeit összeadjuk, ez adja majd a csapatok pontszámát. A futtatott kódok szabadon módosíthatók a játék során bármikor, azonban a sportszerűség érdekében a következőket tartásuk be:

- csak 100% Java kódot futtassatok (natív hívások, egyéb külső programok használata nem engedélyezett, de például BeanShell vagy Nashorn szkriptek igen)
- csak a helyszínen lévő eszközeiteket használjátok
- ne akadályozzátok a többiek munkáját (például fölösleges hálózati forgalom generálásával)

² Egy bolygón így vagy nincs csomag, vagy 1 csomag van, vagy 2 csomag van, de akkor az egyik mindenképp gazdátlan.

A program bemutatása

Az elkészült programot egy maximum 5 perces prezentáció keretében kell bemutatni a zsűrinek, illetve a zsűri kérdéseket tehet fel a programmal kapcsolatban.

A prezentációban a következők bemutatását kérjük:

- milyen algoritmusokat / egyedi megoldásokat használtak a hajók irányításához
- milyen technológiákat használtak a program elkészítéséhez
- milyen nehézségek merültek fel a feladat megoldása során, és hogy oldották meg

A prezentáció bemutatásán csak a csapat tagjai és a zsűri vesz részt, a többi versenyző csapat nem.

A prezentációkat a döntőn is lesz még lehetőségek módosítani, előzetesen csak formai átnézés miatt kértük be.

Java Challenge szerver REST API interfész specifikáció v1.5

Változások az 1.4 verzió óta

- *wherels* válaszban a *pack* mező nem tömb
- *dropPackage* válaszban *ALREADY_OCCUPIED* státusz is lehet

Dokumentum célja

Ez a dokumentum tartalmazza a döntőben használható REST API interfész leírását.

Az interfész használata

Az egyes API hívásokat kizárólag http(s) protokollon keresztül lehet elérni, POST és GET metódusok használatával. Ha a játék még nem indult el vagy már véget ért, a szerver minden kérésre 503-as hibával válaszol.

A szerver túlterhelésének elkerülése érdekében minden kérés között (felhasználónként) legalább 100 ms időnek kell eltelnie. Ellenkező esetben az adott kérés 403-as hibát okoz.

Ha a hajó aknára futott és a javítási idejét tölti, bármilyen kérés, amihez a hajó használata szükséges, 423-as hibát dob.

GET kérések esetén, kliens oldalon érdemes kikapcsolni a cache-elést az esetleges ebből adódó hibák elkerülése érdekében. Alapértelmezett karakterkódolás az UTF-8.

Authentikáció

A felhasználók azonosítását [http basic authentication](#) segítségével végezzük. Ennek megfelelően minden kérés fejlécében szerepelnie kell az Authorization mezőnek. Pl.:

```
GET /JavaChallenge/rest/ping HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: hu-hu,hu;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: Basic dHN0dXNlcjpwZWxzem8=
Connection: keep-alive
Cache-Control: max-age=0
```

A felhasználónévben nem lehet ékezetes karakter.

Szolgáltatások

Elérés:	/JavaChallenge2/rest/ ping
Metódus:	GET
Leírás:	Ezzel lehet ellenőrizni a szerver működését.
Kérés paraméterek:	-
Válasz típusa:	text
Válasz leírása:	Normál esetben a válasz: OK

Elérés:	/JavaChallenge2/rest/ getGalaxy	
Metódus:	GET	
Leírás:	Visszaadja a bolygókat és a bolygókon lévő csomagokat.	
Kérés paraméterek:	-	
Válasz típusa:	json	
Válasz leírása:	<pre> { "planets": [<planets>] (List) } <planet>: { "name": "Amazonia", (String) "x": 63.0, (double) "y": 6.0, (double) "packages": [<packages>] (List) } <package>: { packageId: 1370, (int) lastPlanet: "Amazonia" (String) lastOwner: null (String) } </pre>	
Válasz mezők jelentése:	planets:	az összes bolygót tartalmazó lista
	name:	a bolygó neve (egyedi azonosítója)
	packages:	az adott bolygón lévő csomagok listája
	packageId:	csomagazonosító, később ennek segítségével hivatkozunk a konkrét csomagra
	lastPlanet:	a csomag utolsó helye
	lastOwner:	a csomag utolsó tulajdonosa

Elérés:	/JavaChallenge2/rest/ go	
Metódus:	POST	
Leírás:	Elindítja a versenyző űrhajóját az adott célállomás felé.	
Kérés paraméterek:	planetName=Amazonia (String) A célállomás (bolygó) neve. shipNum=1 (Integer) A hajó sorszáma (0-2). Ha nincs megadva értéke 0 lesz.	
Válasz típusa:	json	
Válasz leírása:	<pre> { status: "MOVING", (enum) arriveAfterMs: 46000, (Integer) destination: "Tarantulon 6" (String) } </pre>	
Válasz mezők jelentése:	status:	a parancs eredményét adja meg. Értéke lehet: <ul style="list-style-type: none"> ALREADY_MOVING – az űrhajó már most is mozgásban van, először meg kell várni az előző mozgás végét NOTHING_TO_DO – az űrhajó már most is a megadott célállomáson van UNKNOWN_PLANET – ismeretlen bolygónév MOVING – az űrhajó elindult a megadott cél felé
	arriveAfterMs:	hány ms múlva ér a célhoz az űrhajó (csak akkor nem null, ha a status MOVING)

	destination:	az űrhajó új célállomása (csak akkor nem null, ha a status MOVING)
--	--------------	--

Elérés:	/JavaChallenge2/rest/ wherels	
Metódus:	GET	
Leírás:	Megadja az űrhajók aktuális helyét, úti célját, szállított csomagjait.	
Kérés paraméterek:	–	
Válasz típusa:	json	
Válasz leírása:	<pre>{ "userName" : "testuser", "ships" : [{ "planetName" : "Beta Niobe", "targetPlanetName" : "Beta Niobe", "arriveAfterMs" : null, "pack" : <package>, "shipNum" : 0 }, { "planetName" : "Beta Niobe", "targetPlanetName" : "Beta Niobe", "arriveAfterMs" : null, "pack" : <package>, "shipNum" : 1 }, { "planetName" : "Beta Niobe", "targetPlanetName" : "Beta Niobe", "arriveAfterMs" : null, "pack" : <package>, "shipNum" : 2 }], "remainingMines" : 0 }</pre>	
Válasz mezők jelentése:	userName:	Felhasználónév.
	planetName:	Az a bolygó, ahol a hajó épp van. Ha az űrhajó mozgásban van, akkor null.
	targetPlanetName:	aktuális célállomás (ha mozgásban van)
	arriveAfterMs:	hány ms múlva ér a célhoz az űrhajó (ha mozgásban van)
	pack:	az űrhajón lévő csomag, lehet null
	shipNum:	az űrhajó sorszáma
	remainingMines:	hány telepíthető akna van még

Elérés:	/JavaChallenge2/rest/ whereAre	
Metódus:	GET	
Leírás:	Megadja az összes játékos űrhajójának aktuális helyét, úti célját	
Kérés paraméterek:	–	
Válasz típusa:	json	
Válasz leírása:	<pre>{ "userPositions" : [{ "userName" : null, "ships" : [{ "planetName" : "Aldebaran", "targetPlanetName" : "Aldebaran", "arriveAfterMs" : null, "pack" : null, "shipNum" : 0 }, { "planetName" : "Aldea", "targetPlanetName" : "Aldea", "arriveAfterMs" : null, "pack" : null, "shipNum" : 1 }, { "planetName" : "Aldebaran", "targetPlanetName" : "Aldebaran", "arriveAfterMs" : null, "pack" : null, "shipNum" : 2 }], "remainingMines" : null }, ...]</pre>	
Válasz mezők jelentése:	userName:	Felhasználónév.
	planetName:	Az a bolygó, ahol a hajó épp van. Ha az űrhajó mozgásban van, akkor null.
	targetPlanetName:	aktuális célállomás (ha mozgásban van)
	arriveAfterMs:	hány ms múlva ér a célhoz az űrhajó (ha mozgásban van)
	pack:	mindig null
	shipNum:	az űrhajó sorszáma
	remainingMines:	mindig null

Elérés:		/JavaChallenge2/rest/ pickPackage	
Metódus:		POST	
Leírás:		Felveszi a megadott csomagot az űrhajóra.	
Kérés paraméterek:		packageId=995 (int) A felvenni kívánt csomag csomagazonosítója. shipNum=1 (Integer) A hajó sorszáma (0-2). Ha nincs megadva, értéke 0 lesz.	
Válasz típusa:		json	
Válasz leírása:		<pre>{ status: "PACKAGE_PICKED" (enum) remainingCapacity: 0 (Integer) }</pre>	
Válasz mezők jelentése:		status:	a parancs eredményét adja meg. Értéke lehet: <ul style="list-style-type: none"> PACKAGE_PICKED – a csomagot sikeresen felvette NOT_FOUND – a megadott csomag nem található LIMIT_EXCEEDED – a csomag már nem fér be az űrhajóba (túl sok csomag van már rajta) USER_NOT_ON_THE_PLANET – az űrhajó nem azon a bolygón van, ahol a csomag
		remainingCapacity:	a csomag felvétele után még ennyi csomagnak van hely a hajón

Elérés:	/JavaChallenge2/rest/ dropPackage		
Metódus:	POST		
Leírás:	Lepakolja a csomagot a hajóról.		
Kérés paraméterek:	shipNum=1 (Integer) A hajó sorszáma (0-2). Ha nincs megadva, értéke 0 lesz.		
Válasz típusa:	json		
Válasz leírása:	<pre>{ status: "PACKAGE_DROPPED", (enum) scoreIncrease: 0 (int) }</pre>		
Válasz mezők jelentése:	status:	a parancs eredményét adja meg. Értéke lehet: <ul style="list-style-type: none"> PACKAGE_DROPPED – a csomagot sikeresen letette NOT_WITH_USER – nincs csomag a versenyző hajóján NOT_AT_DESTINATION – az űrhajó még nem érkezett meg a bolygóhoz ALREADY_OCCUPIED – a bolygót már elfoglalták, nem rakható le a csomag 	
	scoreIncrease:	hány pontot kapott a felhasználó a fuvarért	

Elérés:	/JavaChallenge2/rest/installMine	
Metódus:	POST	
Leírás:	Telepít egy aknát.	
Kérés paraméterek:	planetName=Aldea (String) A bolygó neve, ahova az aknát telepítjük. shipNum=1 (Integer) A hajó sorszáma (0-2). Ha nincs megadva, értéke 0 lesz.	
Válasz típusa:	json	
Válasz leírása:	<pre>{ status: "INSTALLED", remaining: 9 }</pre>	
Válasz mezők jelentése:	status:	a parancs eredményét adja meg. Értéke lehet: <ul style="list-style-type: none"> INSTALLED – az aknát sikeresen telepítette USER_NOT_ON_THE_PLANET – a hajó nem az adott bolygón van PLANET_NOT_OWNED – a bolygó nincs a játékos tulajdonában LIMIT_EXCEEDED – elfogyott az akna
	remaining:	hány aknánk maradt