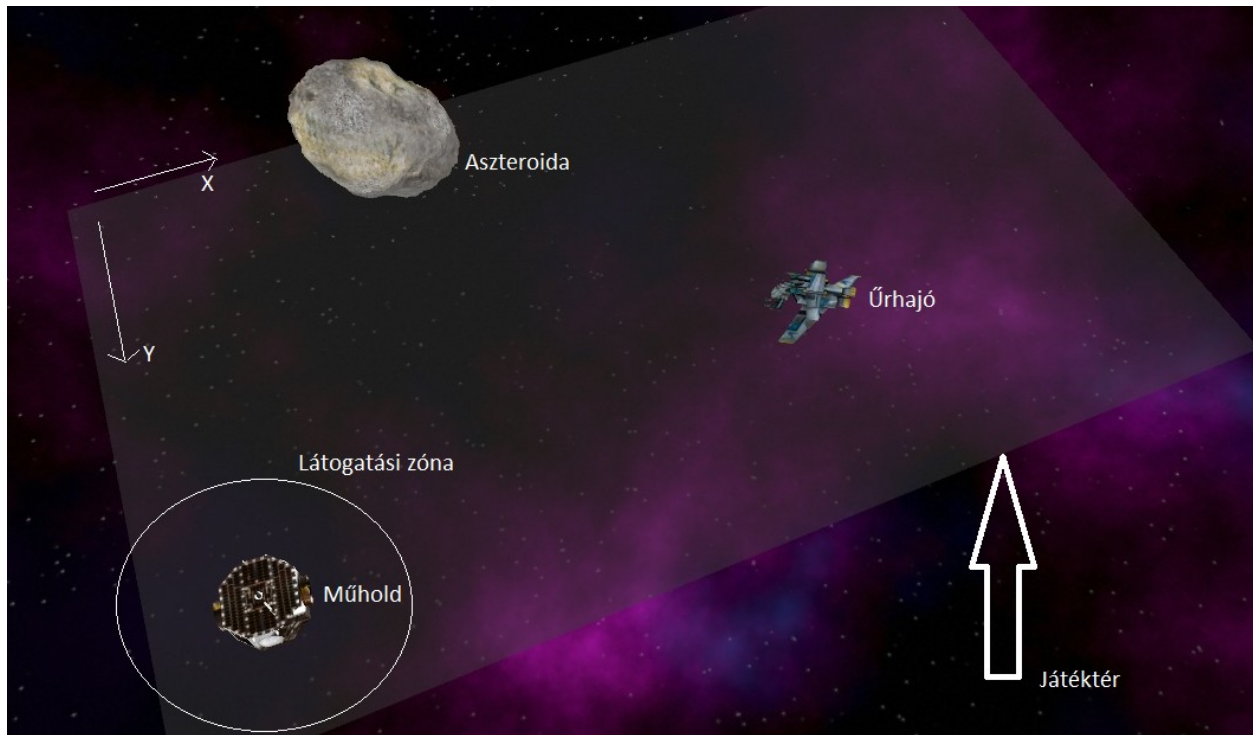


Űrhajós játék



Összefoglaló a feladatról

Adott egy űrhajós játék. A feladatokat lesz irányítani az űrhajót. Egy ágenszt kell tehát implementálni.

Részletes leírás

Idő

A játérendszer szimulált környezetben működik, a valós időt diszkrét időpillanatokkal (tick) helyettesítjük.

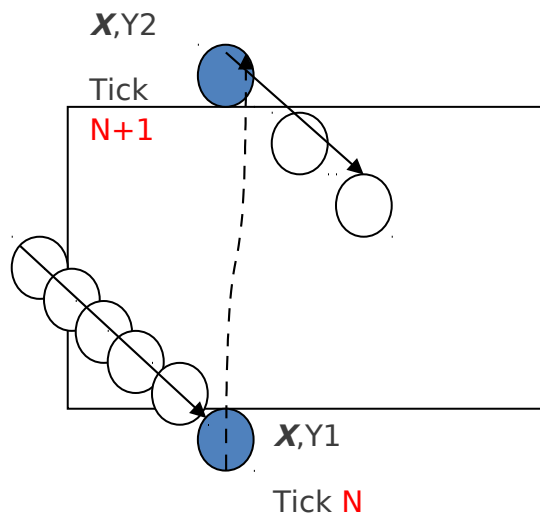
Játéktér

A játék 2 dimenziós, mindent 2 dimenziós vektorokkal jellemzünk. Az X irányú koordináták balról jobbra növekednek, az Y irányú koordináták fentről lefelé.

A játéktér mérete véges, ezt egy vektor írja le. Pl.: $\{100.0, 200.0\}$. A játéktérben 3-féle objektum létezhet. Mindegyik kör alakú, egy pozíció vektor (position) és egy sugár (radius) jellemzi őket. Az objektumok:

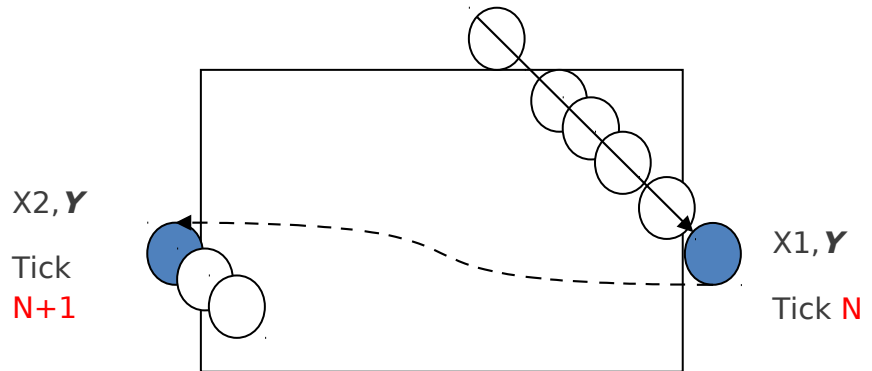
- Műhold

- Nem végez mozgást, egy helyben áll.
 - Jellemzi a látogatási zóna (visitationRadius), melynek sugara nagyobb, mint a műhold sugara
 - Az összes műhold az első tick előtt már a játéktérben van. Új műhold nem jelenik meg játék közben.
 - Kezdetben minden műhold *elromlott* állapotban van. Ha az űrhajó széle a műhold látogatási zónáján belülre kerül, akkor úgy tekintjük, hogy a játékos megjavította az adott műholdat.
- Aszteroida
 - Egyenes vonalú egyenletes mozgást végez. Sebességét egy vektor írja le.
 - Amennyiben a játéktérrel elhagyja az aszteroida teljesen, akkor a pálya átellenes pontjában tér vissza a játékba.
- Példa 1.



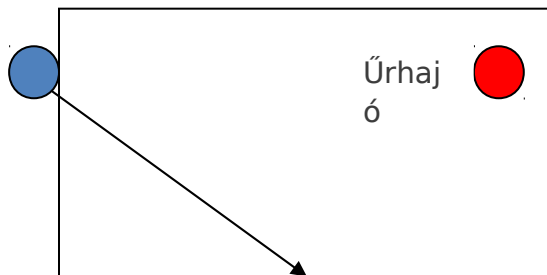
A fenti esetben jól látszik, hogy amikor visszatér az aszteroida, az X koordinátája nem változik.

Példa 2.



A fenti esetben jól látszik, hogy amikor visszatér az aszteroida, az Y koordinátája nem változik.

- A játék során egy aszteroida bármikor (bármely tickben) megjelenhet a játéktér szélén.



- Űrhajó

- Jellemzi egy szög, mely azt írja le, hogy milyen irányba néz az űrhajó eleje. Ez a szög módosítható a fordulás paranccsal. (lásd lejjebb)

- A szög radiánban értendő. Az űrhajó iránya így jellemezhető tehát:

- $0 \rightarrow \{x=1, y=0\}$
- $\pi/2 \rightarrow \{x=0, y=1\}$
- $\pi \rightarrow \{x=-1, y=0\}$
- $3\pi/2 \rightarrow \{x=0, y=-1\}$

Általánosabban:

Az irányvektort így kaphatjátok meg:
 $\{x=\cos(a), y=\sin(a)\}$, ahol a jelöli az aktuális szöget.

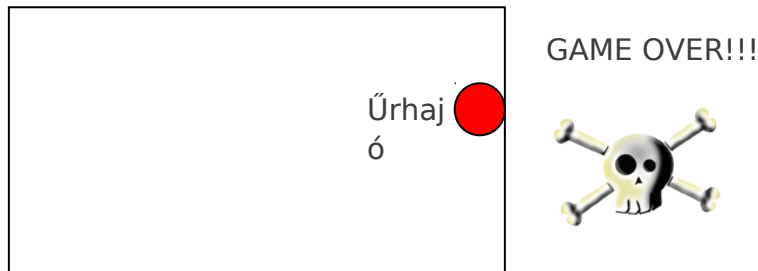
- Amennyiben nincs bekapcsolva a hajtómű egyenes vonalú egyenletes mozgást végez. Sebességét egy vektor írja le. A játék elején a sebessége 0 (nullvektor). A sebesség módosítható a hajtómű bekapcsolásával azaz gyorsulás paranccsal. (lásd lejjebb).
- $\{x=\cos(a), y=\sin(a)\}$, ahol a jelöli az aktuális szöget. Az űrhajó nem látja a teljes játékteret, mert radarja csak adott távolságon belül lévő objektumokat képes detektálni (fogOfWarRadius). Egy objektumot akkor lát az űrhajó, ha középpontjuk távolsága kisebb mint fogOfWarRadius és az objektum sugarának összege.

Aszteroidák és műholdak nem tűnnek el a játék során.

Egyszerűsítésképpen, az aszteroidák semmilyen kölcsönhatásba nem lépnek a műholdakkal. Tehát nem fog felrobbanni egy műhold, ha egy aszteroida nekiütközik: Ilyenkor egyszerűen csak átmennek egymáson.

A játék célja és a pontozás

A játék célja, hogy időkorláton belül az űrhajó az összes műholdat meglátogassa, miközben nem ütközik aszteroidával, műholddal, vagy a pálya szélével:



A játék akkor ér véget, ha az űrhajó megsemmisül, vagy ha az előre megadott időkorlát lejár. (Illetve ha a processz nem válaszol megadott időn belül, de erről majd később olvashattok.)

Jutalom jár azért, ha a játékos megjavítja az egyes műholdakat. Ehhez az kell, hogy az űrhajó széle a műhold látogatási zónáján belülre kerüljön.

A pontozás a következő prioritási sorrend szerint történik (fontosság szerinti csökkenő sorrendben):

1. Műhold látogatások száma

2. Utolsó műhold látogatás ideje
3. Űrhajó megsemmisülésének ideje

Tehát a pontszámot úgy lehet maximalizálni, ha a játék során az Űrhajó az összes műholdat meglátogatta, ezt minél gyorsabban tette, és a játékidő lejártáig életben marad.

Irányítás

Minden egyes időpillanatban (tick-ben) egyetlen egy parancs adható ki.

A parancsok:

- Gyorsulás (ACCELERATE). A gyorsulás abba az irányba történik, amelybe az Űrhajó éppen néz. A gyorsítás parancs csak egy tickig érvényes. Több ticken át tartó gyorsításhoz minden tickben ilyen parancsot kell küldeni. A gyorsulás mértéke játékonként konstans. Ez a konstans legyen A (=spaceShipAcceleration). Az elérhető sebességnek is van maximuma, ezt jelöljük M -mel (=spaceShipMaximumSpeed). A gyorsítás a következő algoritmus szerint működik:

```
Vector v := az Űrhajó sebességvektora  
Vector a := az Űrhajó forgásszögével egyező irányú, A hosszúságú vektor  
v := v+a  
if ( v.length() > M )  
    v := v*(M/v.length())
```

- Fordulás (TURN). Forgatja az Űrhajót a tengelye mentén. Megadható paraméterként a szög. A megadott szög relatív, tehát hozzáadódik az eddigi szöghöz.
 - o A paraméterként megadott szög (radiánban) abszolút értéke nem haladhat meg egy maximumális értéket (spaceshipMaximumTurningAngle). Amennyiben mégis meghaladja, akkor egyszerűen a szög előjelétől függően \pm spaceshipMaximumTurningAngle kerül feldolgozásra.
- Üres parancs (IDLE). Az Űrhajó ebben a tickben ne csináljon semmit.

Kommunikáció/Protokoll

A feladat során olyan processzt kell írjatok, mely a külvilággal a standard outputon kommunikál és a külvilágról minden információt a standard inputon szerez meg.

A kommunikáció üzenet alapú és bináris. A kommunikációhoz a google [protobuf](#) library-ét használjuk. Az üzeneteket az alábbi protobuf fájlokkal írjuk le:

- *World.proto*
 - o Minden egyes tickben kap majd a processz egy ilyen üzenetet.
 - o A legelső tickben kapott World üzenetben ki lesz töltve az opcionális gameConfig mező. Az összes többi tickben csak az objects mező fog tárolni adatokat.
 - o A gameConfig mező a következő konstansokat tartalmazza:
 - timeLimit: ennyi tick után jár le az idő és lesz vége mindenképpen a játéknak. Ez az érték legfeljebb 50000 lesz.
 - spaceShipMaximumSpeed: az űrhajó sebessége ezt az értéket nem tudja átlépni (lásd feljebb a parancsoknál). Legalább 1 és legfeljebb 4.
 - spaceshipMaximumTurningAngle: a maximális lehetséges fordulási szög a Turn parancshoz radiánban.
 - size: A téglalap alakú játéktér mérete. Mindkét dimenziója a [100, 50000] intervallumban lesz.
 - fogOfWarRadius: Ilyen messzire lát el az űrhajó. Legalább az űrhajó átmérőjének 20-szorosa.
 - spaceShipAcceleration: egy Acceleration parancsra ilyen mértékű gyorsulást ad le a hajtómű. Legalább 0.1 és legfeljebb 4.
 - o Objects mező csak azokat az objektumokat fogja tartalmazni, amelyek fogOfWarRadiusnál közelebb vannak az űrhajóhoz.
 - o Minden tickben pontosan egy Spaceship objektum lesz az Objects mezőben. Ez az objektum írja le az űrhajó aktuális állapotát.
- *Command.proto*
 - o Minden egyes tickben egy ilyen üzenetet várunk a processztől. Ha nem akar semmit csinálni a hajó, akkor IDLE üzenetet kell küldeni. Ha nem kapunk választ a processztől megadott időkorláton (40ms) belül, vagy ha hibásan kódolt üzenetet kapunk, akkor a játék véget ér.

Segítségképpen mellékelünk egy fájlt, melyet használhatjátok arra, hogy üzeneteket küldjetek illetve fogadjatok: *ProtobufUtil.hpp*

Példa

A csatolt példában egy buta űrhajóhoz érkező üzenetek sorozatát láthatjátok. A példa processz minden körben a forgatás parancsot adja ki, és megjeleníti a standard erroron az éppen fogadott üzenetet.

- *DummyAgent.cpp*
- *DummyAgent.stderr*
 - o A standard error kimenete a példa processznek.

Fordítás

A fordítás során a kapott cpp és hpp fájlok mellé be fogjuk másolni a proto fájlokat és a *ProtobufUtil.hpp*-t. Ezután a következő parancsokat adjuk ki:

```
1. protoc --cpp_out=. *.proto

2. g++ -std=c++11 -pedantic -static -O2 -isystem $
   (protoc_dir)/include -isystem $(boost_dir)/include -L $
   (protoc_dir)/lib *.cc *.cpp -lprotobuf -lpthread -o spaceship
```

Tehát használhatjátok a BOOST libraryk header-only részét is.

Az általunk használt protoc verziója: 2.4.1,
a gcc verziója 4.7.1,
a boost verziója 1.51.

Tesztelés

A program stderr streamre írhat bármilyen információt. Ez a kimenet játékonként nem haladhatja meg az 5Mbyte-ot. A feladatok tesztelése után az összes játék stderr kimenetét letölthetitek az internetes felületről. (megjegyzés: az óránkénti tesztelés során az ágenseteket nem azokkal a tesztesetekkel futtatjuk, mint amelyek majd a végső pontozást meghatározzák)

3 pályával fogunk tesztelni.

1. Teljesen üres pálya, hogy tesztelhessétek a hajó mozgását.
2. Egy műholdat tartamazó pálya. A műhold a pálya közepén lesz.
3. Ugyanaz mint a 2. pálya, csak még plusz egy aszteroida is megjelenik rögtön az 1. tickben.

A részletesebb információk a protobuf üzenetekből derülnek ki.

Sok sikert mindenkinek!

may the force be with your spaceship