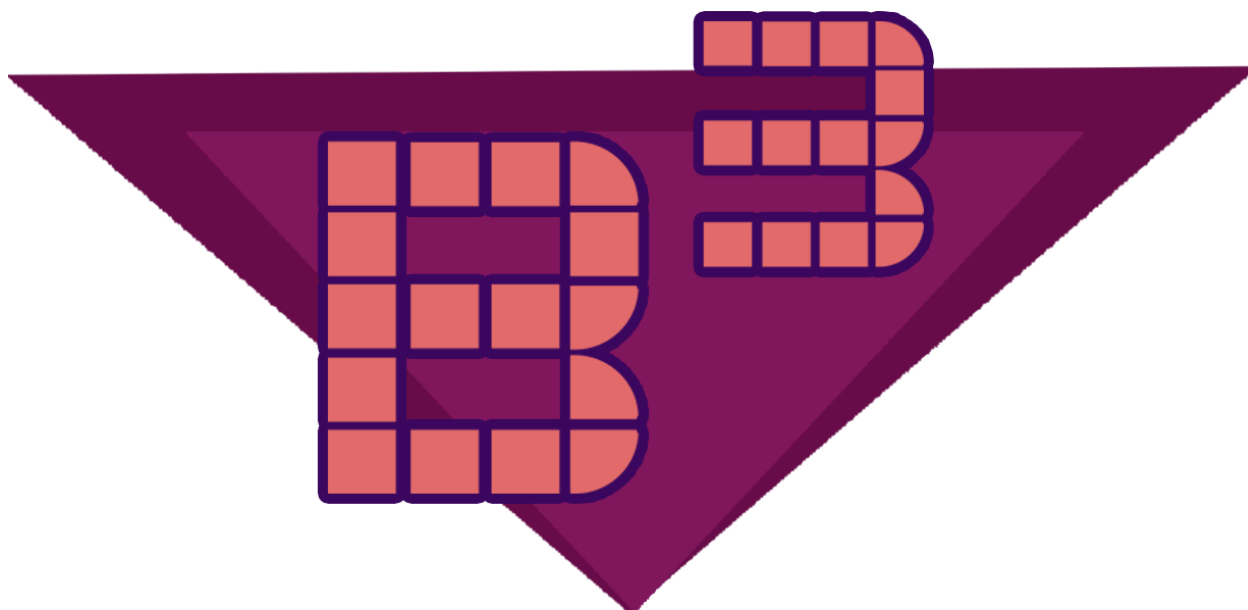




Bakonyi Bitfaragó Bajnokság
Döntő forduló
2025. November 22.
10:00-17:00

Kedves Bitfaragók!

Köszöntünk a verseny döntő fordulóján. A mai döntő során kommunikálni kell majd egy szerverrel, és megoldani az általa adott feladatokat. Ezen felül fontos lesz egy szép honlap elkészítése az eredmények prezentálásához.

A téma

Az előző fordulóknak során is segítettünk párszor az űrt felfedező expedíciónak. Most is ezt kell tennünk. Ezúttal a galaxis egy szegletébe vándorolt egy kirendelt expedíció, ahol értékes ércet tartalmazó aszteroidák vannak. A feladatuk, hogy ebből minél többet kibányásszanak, és vissza szállítsanak. Erre a feladatra pedig hatéridejük is van, hiszen a magukkal cipelt ellátmány nem tart ki örökké.

Feladatok

A fő feladat a rendelkezésre álló hajók irányítása, fejlesztése, illetve új hajók építése, annak érdekében, hogy az idő leteltével minél több érc legyen a készleten. A feladatok három csoportra oszlanak:

- Az 1-6 feladatok ismertetik a rendszert és a szabályokat. A bennük lévő leírások fontosak, de magukat a feladatokat nem kötelező megoldani és pontot sem érnek (tesztelésre viszont használhatóak).
- A 7-16 feladatok a fő pontszerző források. Mindegyikben a cél ugyanaz: minél több ércel rendelkezni az idő lejártakor. A feladatok között az egyetlen eltérés a pálya. Többször is lehet rájuk megoldást beküldeni, a pontszám növelése érdekében. Minden feladat esetén csak a legjobb pontszám számít, mindegy, hogy mennyi és milyen másik próbálkozás volt még.
- A 17-20 feladatok ugyan olyanok, mint a 7-16 feladatok, azzal az eltéréssel, hogy ezek csak a verseny utolsó 10 percében lesznek majd elérhetőek. Ezek segítségével teszteljük, hogy a többi pályára kifejlesztett algoritmusok mennyire működnek új, ismeretlen pályákon.

Kis feladatok rendszere

A kis feladatok során egy központi rendszerrel kell kommunikálni, amely folyamatosan számon tartja majd az előrehaladást. A kommunikáció távoli kérésekkel történik, JSON formátumú üzenetek küldésével. A feladatok különböző kérdéseket fognak tartalmazni, amit a szerver tesz fel, és neki kell megküldeni a választ. A feladaton belüli több kérdés különböző, véletlenszerű eseteket tartalmaznak a kiterjedt teszteléshez.

A szervert a <https://bitkozpont.mik.uni-pannon.hu/2025/> címen lehet elérni, ide kell majd a kéréseket küldeni. A fordulóra minden csapat külön kap egy azonosító kódot. Ezt a kódot minden, a szervernek küldött üzenethez mellékelni kell majd. Ennek pontos módját a későbbiekben részletezzük. A szerver három típusú kérést tud

kiszolgáltatni, amiket sorban át is beszélünk, de előtte még nézzük a kommunikáció általános módját.

Általános kommunikáció

A kérésekhez POST módszerrel paraméterként kötelező megadni a csapatnak a kódját:

```
{  
    teamcode: „d08055dab5f72a650c0e”  
}
```

A „*teamcode*” paramétert minden üzenethez mellékelni kell, ahol a kód értéke a csapat által külön kapott kód legyen. Ezt kell JSON-be kódolva hozzátenni az üzenethez. Az üzenetben ezen felül más paraméter is lehet.

Például az összes feladat lekérése:

```
xhttp.open("POST", " https://bitkozpont.mik.uni-pannon.hu/2025/gettasks.php", true);  
xhttp.send(JSON.stringify({  
    id: „all”,  
    teamcode: „d08055dab5f72a650c0e”  
}));
```

Minden, a szerver által küldött válaszüzenet egységes felépítéssel rendelkezik:

```
{  
    "status": "success",  
    "data": {...},  
    "message": "",  
    "hash": ""  
}
```

Itt a status a sikerességet jelzi, minden esetben a „success” a jó. A „*data*” a válasz tényleges tartalma, ez lekérésenként változik. A „*message*” különböző üzeneteket tartalmazhat, elsősorban, ha valami nem jó. A „*hash*” néha üres, néha tartalmaz egy kódot, a helyzettől függően.

Minden feladat lekérése

A szerveren a következő címről lehet a feladatok listáját lekérdezni: <https://bitkozpont.mik.uni-pannon.hu/2025/gettasks.php>. A lekéréshez a következő adatok kellenek:

```
{  
    id: „all”,  
    teamcode: „d08055dab5f72a650c0e”  
}
```

Itt az *id* az fixen az „all” értéket kapja.

A válasz üzenetben a „*data*” tartalmazza a feladatok listáját („*task_list*”), és az azok közötti összefüggéseket („*dependencies*”), bár az utóbbi minden esetben üres lesz. A szervertől kapott listában a csapat láthatja azt is, hogy melyik feladatok készítették már el, és melyikre mennyi pontot kaptak.

A feladatok állapota 3 féle lehet:

- OPEN: Ezek a nyitott feladatok, amiken a csapat dolgozhat, de megoldás még nem született.
- COMPLETED: Ezek a már sikeresen megoldott feladatok. A feladat ettől még megoldható újra, különösen, ha a pontszáma növelhető.
- LOCKED: Olyan feladatok, amelyek valami miatt még nem elérhetőek (pl. időkorlát miatt).

Egy feladat lekérése:

Egy feladat részletei is a <https://bitkozpont.mik.uni-pannon.hu/2025/gettasks.php> címen érhetőek el. A lekéréshez a következő adatok kellnek:

```
{
    id: 1,
    teamcode: „d08055dab5f72a650c0e”
}
```

Itt az *id* a konkrét feladat azonosítója legyen. Válaszul a szerver közli az adatait:

```
{
  "status": "success",
  "data": {
    "attempt": 18,
    "description": "descriptions/t01_30291043.html",
    "questions": [
      {
        "ID": "1",
        "question_id": "1",
        "question_type": "COMP",
        "order": "1",
        "params": {
          "number1": 1155,
          "number2": 3800,
          "type": "ADDITION"
        }
      },
      {
        "ID": "2",
        "question_id": "1",
        "question_type": "COMP",
        "order": "2",
        "params": {
          "number1": 977,
          "number2": 9218,
          "type": "ADDITION"
        }
      }
    ],
    "ID": 1,
    "level": 1,
    "stage": 1,
    "points": 0,
    "state": "COMPLETED"
  },
}
```

```

"message": "",
"hash": "2a937a6922744436e9bfb911e123e585c0b4bf96"
}

```

Itt a fontos rész a „data”-n belül van. Az „attempt” csak azt tartja számon, hogy hányszor kérte le a csapat ezt a feladatot, ez csak a szervernek fontos. Egyrészt fontos a „description” által tartalmazott leírás. Ez egy HTML oldal címét tartalmazza a szerveren (bár a kiterjesztés néha php), amit célszerű külön megnyitni, hiszen ez fogja mindig tárolni az adott feladathoz tartozó részletes leírást. A „questions” rész sorolja fel egy tömbben az egyes kérdéseket, amire fentebb látható egy példa. A kérdésnek van két azonosítója (ebből az ID kell), egy típusa, és egy sorrendje a feladaton belül (ez a két adat nem szükséges most). A kérdéshez tartozó „params” a kérdés típusát tartalmazza, valamint egyéb paramétereket, amelyek részleteit a kérdések leírása tartalmazza majd. A „hash” adat fontos, mert ezt majd a szervernek vissza kell küldeni.

A feladat leírását tartalmazó html fájl bizonyos feladatoknál tartalmazhat extra információt a feladatról, vagy példát a szerver által küldött kérdésre, és a megfelelő válaszra. Ezek nyilván arra vannak, hogy a kérdés és válasz felépítését egyértelműsítsék. A későbbi feladatok esetén már nincs ilyen, addigra már ismerős lesz a rendszer.

Válaszok elküldése

Amikor a feladat megoldásra került (minden kérdésre van válasz), akkor a válaszokat el kell küldeni a szervernek. A megoldásokra időkorlát nincs, és többször is megpróbálható ugyanazokkal a paraméterekkel. Ha kész a megoldás, az eredményt el kell küldeni a szervernek.

A válasz beküldéséhez a <https://bitkozpont.mik.uni-pannon.hu/2025/answer.php> címre kell kérést küldeni a következő paraméterekkel:

```

{
    id: 1,
    teamcode: „d08055dab5f72a650c0e”,
    original_data: {...},
    original_hash: „...”,
    answer_data: [...]
}

```

Itt az „id” a feladat azonosítója, a „teamcode” a csapat kódja. Az üzenetbe „original_data” néven bele kell tenni a feladat lekérése során megkapott teljes „data” részt, valamint „original_hash” néven a megkapott eredeti „hash” szöveget. Ezek nélkül a szerver nem fogja tudni megfelelően értékelni a válaszokat. A tényleges válaszok az „answer_data” részben vannak, ami egy tömbben tárol egy-egy választ minden kérdéshez, ilyen módon:

```

{
    id: ...,
    answer: ...
}

```

Az „id” a kérdés azonosítója (ami a kérdés leírásában az „ID”), az „answer” a válasz.

A fenti példa feladatra egy példa válasz üzenet tehát így néz ki (a megoldás ténylegesen helyes az előző példához, de a véletlenszerűen generált paraméterek miatt úgysem használható fel egy az egyben):

```
{
  "id": "1",
  "teamcode": "dc6936418b366236c4cf",
  "original_data": {
    "attempt": 18,
    "description": "descriptions/t01_30291043.html",
    "questions": [
      {
        "ID": "1",
        "question_id": "1",
        "question_type": "COMP",
        "order": "1",
        "params": {
          "number1": 1155,
          "number2": 3800,
          "type": "ADDITION"
        }
      },
      {
        "ID": "2",
        "question_id": "1",
        "question_type": "COMP",
        "order": "2",
        "params": {
          "number1": 977,
          "number2": 9218,
          "type": "ADDITION"
        }
      }
    ],
    "ID": 1,
    "level": 1,
    "stage": 1,
    "points": 0,
    "state": "COMPLETED"
  },
  "original_hash": "2a937a6922744436e9bfb911e123e585c0b4bf96",
  "answer_data": [
    {
      "id": "1",
      "answer": 4955
    },
    {
      "id": "3",
      "answer": 10195
    }
  ]
}
```

A válaszokra a szerver egy üzenettel reagál, amiben leírja, hogy jónak találja-e a küldött eredményeket. Ebben a válaszban minden feladathoz van egy „*feedback*” mező is, amelyben a szerver speciális információkat közölhet az egyes feladatokról, valamint, amennyiben sikeres volt a megoldás és a pontszám többféle is lehet, egy „*pointsEarned*” mezőben elküldi a feladatra elért pontszámot is.

Vizualizációs feladat

A fő feladatok mellett extra pontok szerezhetőek vizuális megjelenítés elkészítésével. A megjelenítés feladata a pálya és a hajók megjelenítése. Ezen keresztül végig lehet nézni az utasítás sorozat végrehajtását. Egyrészt mutatni kell az adott utasítás sorozat végigfutását, vagyis, hogy mi történik az utasítások hatására. Másrészt hasznos lehet még tovább gondolni, és esetleg lehetővé tenni a kézi játékot, ahol egy felhasználó ezen a felületen tudja a parancsokat megadni. Ennek az értékelését a zsűri fogja végezni. A verseny utolsó 1,5 órájában lehetőség lesz az elkészült megjelenítést bemutatni a zsűrinek (akár többször is, ha még sikerül menet közben javítani rajta). A zsűri a felületet értékelni fogja teljesség, megjelenés, ötletesség, használhatóság szempontjából.

A verseny eredménye

A verseny során a kapott pontszám nagy részét a sikeresen megoldott feladatok teszik ki, valamint a vizuális megjelenítésre is lehet szép számmal extra pontot kapni, azonban az értékelés nem csak ennyiből áll. A verseny végén minden csapatnak tartani kell egy rövid prezentációt, amiben ismertetik a megoldásukat, a kihívásokat, a legjobb eredményeket. Ezt a prezentációt is értékeljük, valamint figyelembe vesszük az elkészített munka pontozásánál.