

# Útikalauz Kódolóknak

---

*Segédlet a Kódolás Órájához*

*Kertész Gábor*

*kertesz.gabor@nik.uni-obuda.hu*

## Tartalom

1. Ne pánikolj! .....	1
2. Star Wars – Galaxis építése kódolással .....	2
3. Minecraft .....	4
4. Kódolás Annával és Elzával .....	6
5. Angry Birds, Plants vs Zombies, Motkány .....	7
6. Flappy Bird .....	8
7. Grok Learning – zászlórajzolás .....	9
a. Sűgő a feladatokhoz .....	10
8. Code Combat .....	11
9. CodinGame .....	13
a. Onboarding .....	16
b. Power of Thor .....	17
c. The Descent .....	18
d. Skynet: The Chasm .....	19

## 1. Ne pánikolj!

A kódolás nem atomfizika, a kód pedig nem hieroglifákból áll: könnyen és gyorsan tanulható az alapja. Döntsd el, hogy mivel szeretnéd kezdeni! A kezdőknek javasolt a Star Wars galaxis kódolása, vagy a Minecraft, de ha profi vagy, bemelegítésnek akkor is jó lehet!

Mi történik ha elakadsz a megoldással?

### 1. szabály: ne pánikolj!

Nyugi, előfordul ez még a legjobbakkal is. Ha pánikolsz, leblokkolsz, és többé már nem a problémamegoldásra koncentrálsz.

Persze mindez nem visz előre. Tudod mi segít?

## 2. szabály: kérdezz bátran!

Még a legnagyobbak sem születtek komplett tudással, inkább bátran kérdeztek, ha nem tudtak.

Érdekes megoldás, és nagyvállalatok programozói meglepően gyakran alkalmazzák, hogy közösen dolgoznak: ketten ülnek egy számítógép, és egy probléma felett, egymást segítik a megoldás menetében. Esetleg beszéld meg a melletted ülővel, hogy dolgozzatok össze!

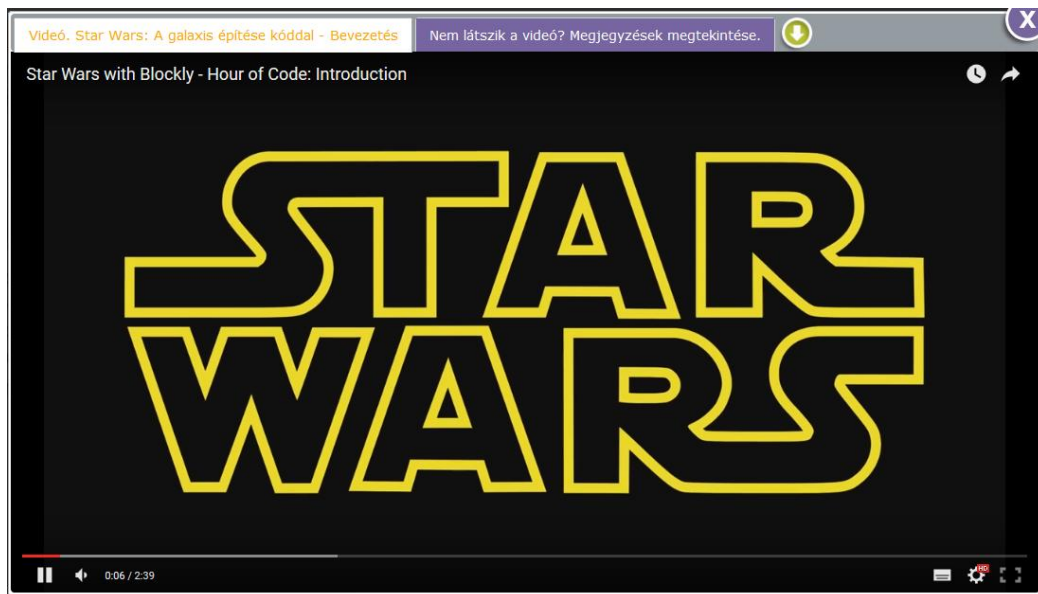
Ha egyéb zűr van, vagy csak félsz oldalra nézni, szólj nyugodtan a teremben levő oktatónak: örömmel fog segíteni.

## 3. szabály: légy közösségi!

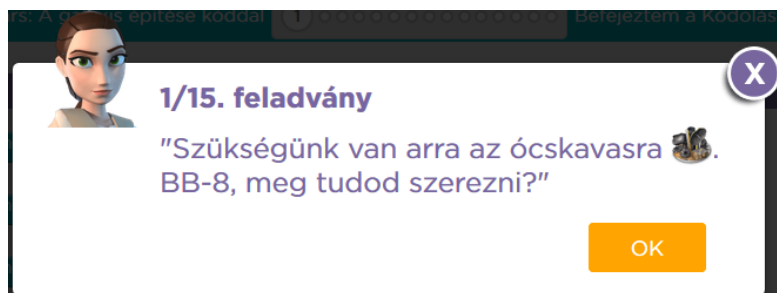
A Facebook világában persze ez nem túl nehéz... De az ösztönzés sosem árt: bátran nyomj a megosztás gombra ott, ahol lehetőséged van rá, mutasd meg ismerőseidnek a játékot amit csináltál (lásd Star Wars vagy Mineecraft utolsó feladványa), vagy mutasd magad és az okleveled, amit a Kódolás Órája végén osztunk ki. A hashtageket szeretjük: #ObudaiEgyetem #KodolasOraja #HourofCode

## 2. Star Wars – Galaxis építése kódolással

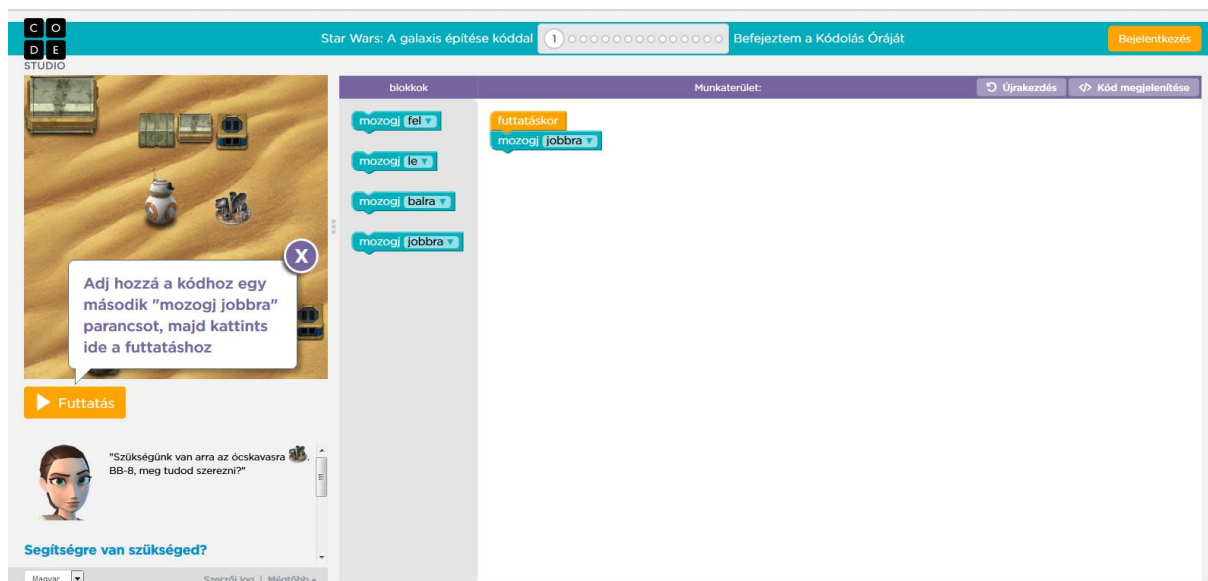
Kezdként megjelenik egy oktatóvideó, amelyet bármikor bezárhatunk, de később újra is nézhetünk. Amennyiben nem jelenik meg magyar felirat, a „Megjegyzések megtekintése” fülre kattintva magyarul leírva olvashatóak az instrukciók.



Az oktatóanyag 15 feladványból áll, melyek sorban egymás után megoldandóak. Minden feladvány előtt megkapjuk a feladat szövegét, majd nekikezdehetünk a kódolásnak!



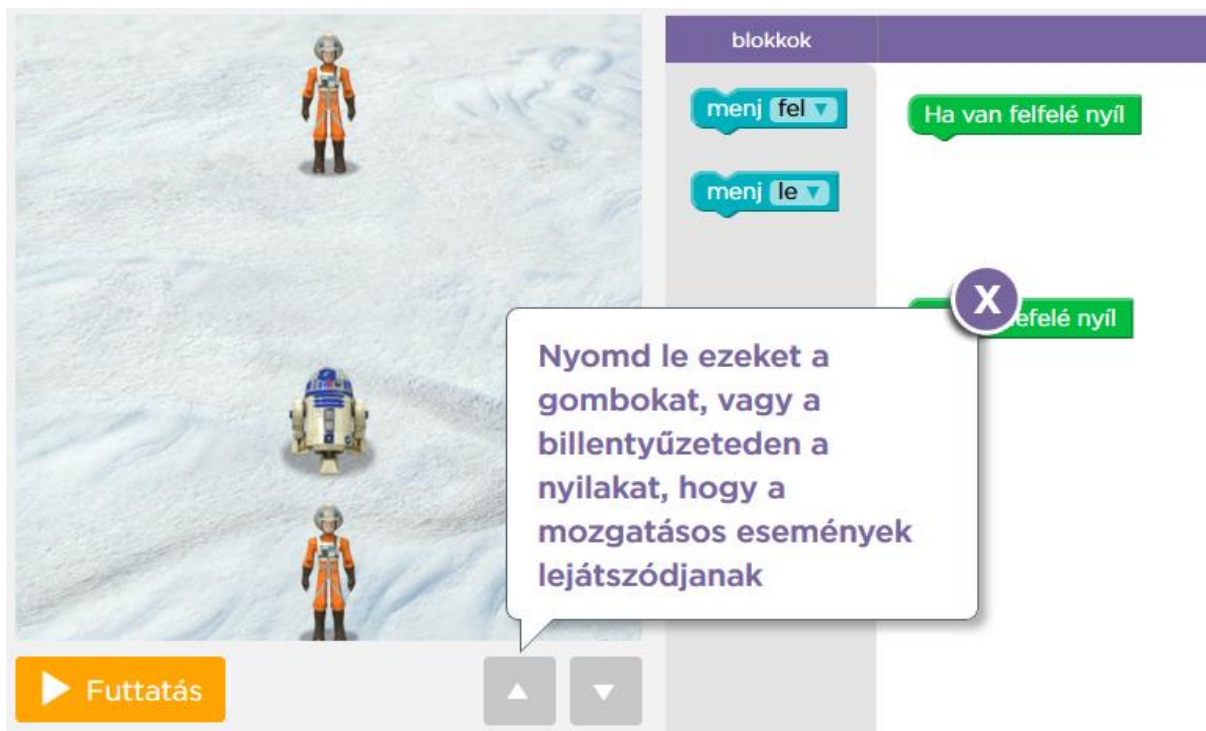
A főablakban balra található pályán tudjuk mozgatni BB-8 droidot, a jobbra található munkaterületen van lehetőségünk kódolni. A blokkok közül egérrel a „húzd és vidd” módszerrel tudunk a munkaterületre blokkokat helyezni, és másikkal összekapcsolni.



Miután sikerült a megfelelő utasításokat kiválasztani, a Futtatás gombra kattintva az összerakott utasítások egymás után végrehajtnak.

A futtatás gomb alatt elérhetjük újra a videót, illetve ha gond lenne a nyelvi beállításokkal, itt tudjuk átállítani magyarra.

A 7. feladványtól kezdődően a baloldali sávban már nem csak a kódolásunk eredményét láthatjuk, hanem egészen interaktív módon egy játékot játszhatunk R2-D2 irányításával. A munkaterületen összeállíthatunk olyan utasításokat, amelyek bizonyos események bekövetkeztekor fognak lefutni. Ilyen esemény lehet a felfelé vagy lefelé mutató kurzorbillentyű lenyomása, melyeket egérekattintással is kiválthatunk.



Végezetül az utolsó feladvány teljesítéséhez saját „játékot” kell készítened: mindent személyessé tehetsz, egyedivé, végül a játékod elmentheted, és megoszthatod ismerőseiddel!

### 3. Minecraft

Hasonlóan a Star Warsnál tapasztaltakhoz, itt is oktatóvideóval segítik a tanulást. Amennyiben esetleg nem lenne magyarul feliratozva a videó, válaszd a „Megjegyzések megtekintése” fület!

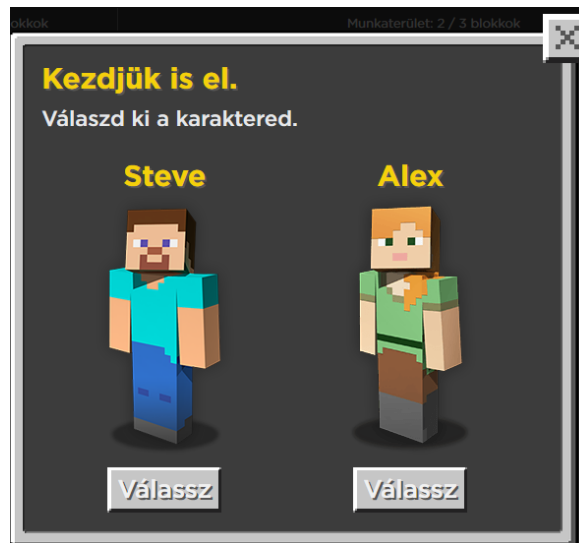
Videó. Minecraft - Bevezetés Nem látszik a videó? [Megjegyzések megtekintése.](#)

Az én nevem Jens Bergensten, de a legtöbben Jebnek hívnak. A Minecraft vezető fejlesztője vagyok itt a Mojangnál Stockholmban. Azt hiszem, tizenegy vagy tizenkét éves voltam, amikor programozni kezdtem, mert játékokat akartam készíteni. Az apukám egyik barátja mondta, hogy ha játékokat akarok csinálni, akkor meg kell tanulnom programozni. Így kezdtem el. Szeretek alkotni és kitalálni a játékok alapjait. Ez az, amit igazán szeretek a Minecraftban.

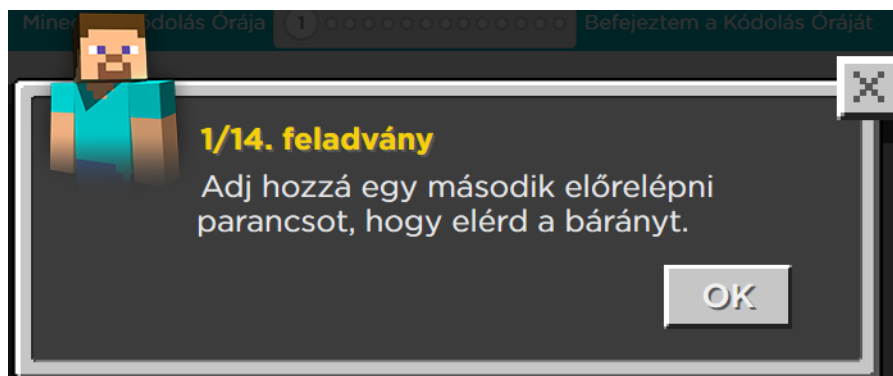
A következő egy órában meg fogod ismerni a számítógép-tudomány alapjait, miközben Alexet vagy Steve-et vezérel, hogy a Minecraft világ egy kis szimulált darbjában mozogjanak.

A programozás hagyományosan szövegesen történik, de ma a Blockly-t fogjuk használni, egy rendszert,

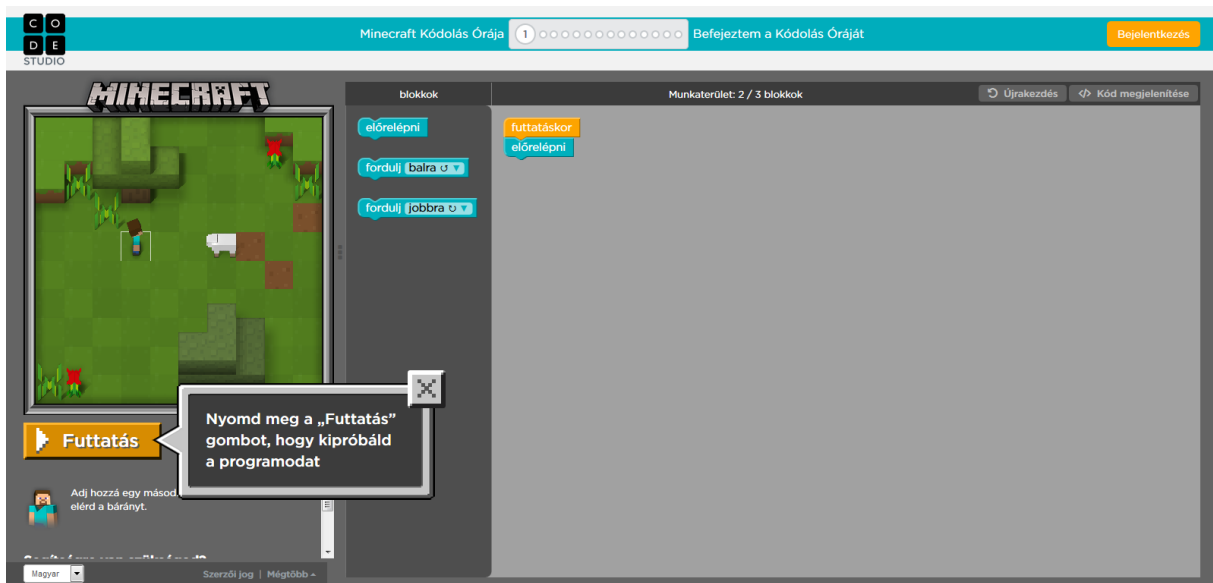
A játék kezdetén választanunk kell egy karaktert, akit irányíthatunk: a fiúknak Steve, a lányoknak Alex.



A Minecraft oktatójáték 14 feladványból áll, melyeket sorban teljesíteni kell: a feladványok sorban követik egymást. Az egyes feladványok rövid feladtleírást tartalmaznak:



A képernyő bal szélén látszik a pálya, rajta a kiválasztott karakterrel, jobbra a munkaterület. A munkaterületen blokkokat kell elhelyezni, és egymással összekötni, így áll össze a kódunk. A kódot összeállítás után a pálya alatti „Futtatás” gombra kattintva tudjuk elindítani. Ha bármilyen nehézségbe ütköznénk, a gomb alatt található a feladat szövege, illetve a videó, melyet bármikor újra megtekinthetünk.



Amennyiben a felület nem magyar nyelvű lenne, a képernyő bal alsó sarkában kiválasztható a magyar nyelv.

Az utolsó feladvány megoldása után saját pályát építhetsz. Hajrá!

#### 4. Kódolás Annával és Elzával

A Frozen-birodalomban játszódó kódolási oktatóanyagban rajzoló utasításokat tudunk kiadni a két korcsolyázó lánynak, amely segítségével különböző formákat lehet kirajzolni a képernyőre. Az játék 20 feladványból áll, és a képernyő felépítése ugyanúgy van kialakítva, mint a Star Wars vagy Minecraft esetén.

Az utolsó szinten saját egyedi rajzot készíthetsz!



C O  
D E  
STUDIO

Kódolás Annával és Elsával

blokkok

menj előre 100 képpontok

fordulj jobbra 90 fok

fordulj balra 90 fok

ugrás előre 100 képpontok

készíts kört  
méret: 10

hozz létre egy négyzet típusú hópelyhet

ismételd 3 alkalommal  
csináld

színbeállítások megadása

színbeállítások megadása Véletlen szín

Futtatás

Hivatalos művész mesterré váltál!  
Készíts egy téli csodaországot.

Magyar Szerzői jog | Még több

## 5. Angry Birds, Plants vs Zombies, Motkány

blokkok

előrelépni

fordulj balra

fordulj jobbra

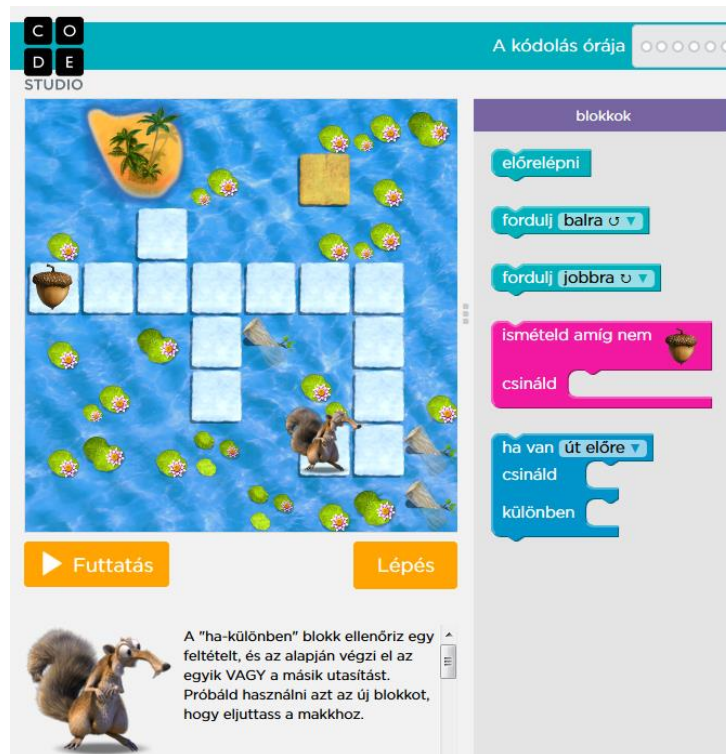
ismételd amíg nem  
csináld

Futtatás

Arghhh - kedves akárci. Én lenni zombi. Zombi lenni éhes. El... kell... a napraforgóhoz... El tudnál vezetni oda csupán 5 utasítással?

Magyar Szerzői jog | Még több

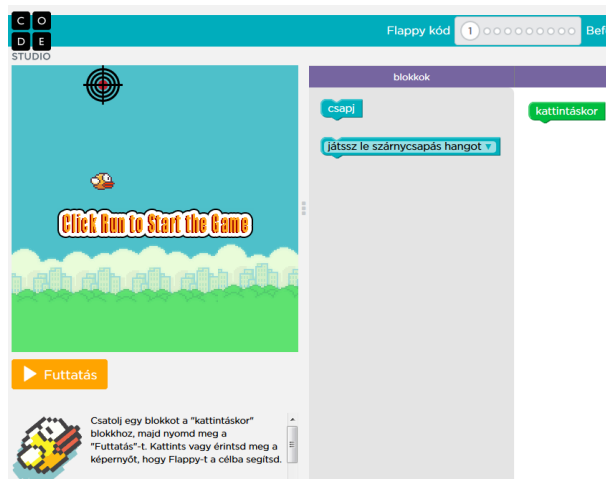
Az eredeti Hour of Code oktatóanyagban előbb a dühös madarat kell a malachoz irányítani, később a zombinak kell utasításokat adni, hogy megtalálja a napraforgót. A tananyag segítségével előbb megismerhetjük az ismétléses vezérlési szerkezetet, majd a feltételes utasításokat, a Jégkorszakból ismert Motkány segítségével.



Hajrá!

## 6. Flappy Bird

A Flappy Bird egy interaktív játék, azaz a számítógép előtt ülő játékos kattintással tud utasítást adni a madárnak. Ilyenkor eseményeket használunk, a kattintás mint esemény bekövetkeztekor kell valaminek történnie.



A felület felépítése hasonló az előzőkéhez, a bal oldali ablakban alatt a futtatás gombra kattintva elindul játék, a képernyőbe kattintva kiváltható a kattintás esemény. Az utolsó, 10. feladvány egy saját játéklógika kialakításáról szól, rengeteg beállítási lehetőséggel.



The image shows a Flappy Bird game interface on the left and a block-based programming editor on the right. The game interface includes a 'Futtatás' (Run) button and instructions in Hungarian. The programming editor has two columns: 'blokkok' (blocks) and 'Munkaterület' (workspace). The 'blokkok' column contains various event and action blocks like 'átlagosat csap', 'játssz le szárnycsapás hangot', 'nőjön eggyel a pontszám', 'játék vége', 'Normál sebesség beállítása', 'Jelenet kiválasztása: Város (nappal)', 'Karakter beállítása: Sárga madár', 'Akadály beállítása: cső', 'Talaj beállítása: Föld', 'rés beállítása: Normál', 'Normál gravitáció beállítása', and 'Pontszám beállítása 0'. The 'Munkaterület' column shows a sequence of blocks: 'futtatáskor' (when run), 'ha becsapódik a talajba' (if hit the ground), 'kattintáskor' (when clicked), 'ha akadálynak ütközik' (if hit an obstacle), and 'ha átjut egy akadályon' (if passed an obstacle).

## 7. Grok Learning – zászlórajzolás

A Grok Learning oldalain zászlórajzolással próbálkozhatunk Python programozási nyelv segítségével. A felület angol nyelvű, de a felépítés elég egyértelmű. A grafikus felületen egy teknősnek adhatunk utasítást, amely mozgásának irányában vonalakat tud rajzolni, és színekkel kitölteni.

The image shows the Grok Learning 'Flying the Flag' tutorial interface. On the left, there is a navigation menu with 'Flags of the World' and 'Hour of Code: Flags of the World'. The main content area has the title 'Flying the Flag' and a welcome message: 'Welcome! In this tutorial we'll be learning to draw flags by putting together different shapes! You'll learn to draw all these flags. Can you recognise any?'. Below this is a grid of flags: the flag of the Netherlands, the flag of Germany, the flag of Switzerland, and the flag of the United Kingdom. A text box says: 'Once you've learned how to draw simple shapes with code, you'll be able to draw your way around the world! Click the arrow to go to the next page.' There is a 'Teachers' section with a note: 'If you sign up for a free account you'll have access to our teacher'. On the right, there is a code editor with the following Python code:

```

program.py
from turtle import *
forward(100)

```

Below the code editor are buttons for 'Run', 'Terminal', and 'Save'. The 'Submissions' section shows a green bar indicating '#1 All tests passed!' and a list of test results:

- Testing that there's a 100 turtle step line. ✓
- Testing that the line is in the right place. ✓
- Testing that there are no extra lines. ✓

A felület bal oldalán találhatóak az instrukciók, illetve a feladatok. Jobbra fent található a munkaterület, az eddigiektől eltérően itt már kódolni, gépelni kell.

Alul egy rövid sávban jelzi a rendszer, hogy a beküldött kód helyes-e.

A bal oldali instrukciók között a kör alakúak oktatóanyagokat tartalmaznak, míg a sarkára állított négyzetek a konkrét megoldandó feladatokat jelzik.

Ha egy ilyen részhez érkezünk, a megfelelő kódot megírva a „Run” gombra kattintva megtekinthetjük a működést. Ha úgy érzed, sikerült helyes megoldást készítened, akkor a „Mark ...” gombra klikkelve, majd a kódot elküldve „Submit” ellenőrzi a rendszer, és jelzi, ha mindent rendben talált.

### a. Súgó a feladatokhoz

A kódablakban a Python nyelv segítségével kell kódokat írunk: a nyelv egy egyedi eleme, hogy a külön utasítások külön sorokba kerülnek, nem szükséges pontosvesszőket írni.

Fontos, hogy a kódblokkok elején levő

```
from turtle import *
```

sort ne töröljük: ez is a nyelv egy egyedi lehetősége, a turtle nevű függvénykönyvtárban vannak definiálva azok a funkciók, amelyeket később a feladatmegoldáshoz meg fogunk hívni.

Az első feladat, hogy a teknős 100 pixelt előre mozogjon, ehhez a forward() utasítást kell használni, a zárójelek közé írva azt a számértéket, amennyivel szeretnénk mozgatni a teknőst.

A második feladvány összetettebb: a teknősnek egy 100 pixel oldalhosszúságú négyzetet kell kirajzolnia. A megoldáshoz használjuk a right() utasítást, amely a teknőst az óramutató járásával megegyező irányba forgatja, a zárójelben megadott érték pedig az a fokszám mértékben.

A harmadik feladvány során megtanuljuk, hogyan lehet színezn: a color() függvényt alkalmazva lehet színt váltani. A régi líbiai zászló „erdő zöld” vagyis forestgreen színű, és 150 pixel széles, 80 pixel magas. A színnel való kitöltéshez a teknős mozgatása előtt a begin\_fill() utasítást kell kiadni, melyet a mozgás befejeztével az end\_fill() utasítással kell lezárni.

A későbbi feladványokban a left() utasítás is használható a balra forduláshoz, illetve kitöltési rövidítések is alkalmazhatóak.

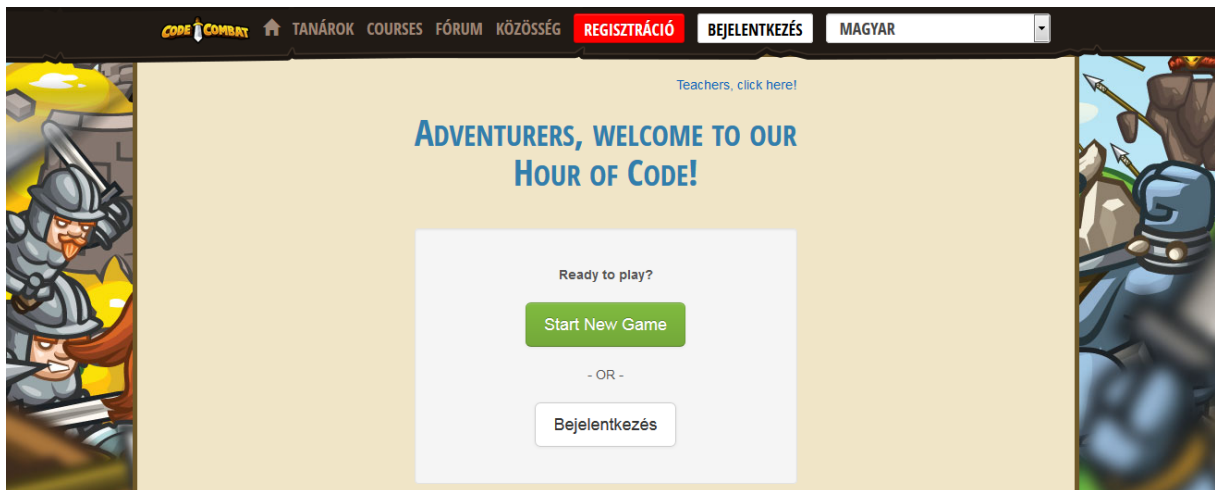


Hajrá!

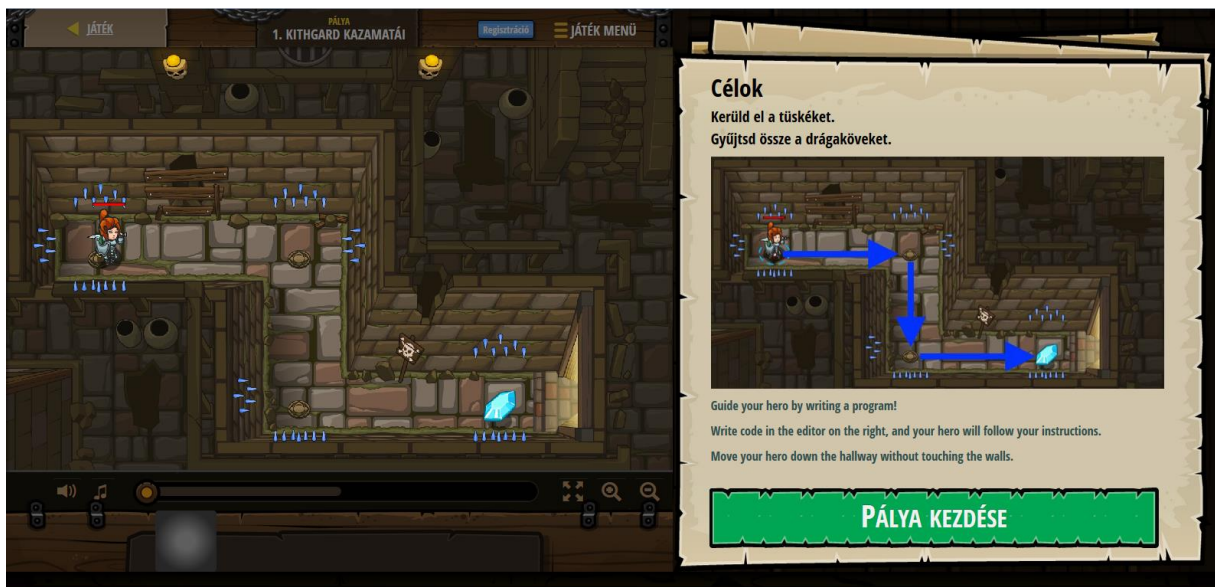
## 8. Code Combat

A CodeCombat egy kifejezetten programozást tanulóknak készített oldal, ahol az egyszerűtől az egészen összetett feladatok megoldásáig írhatunk kódokat.

Az oldal használatához alapesetben regisztrálni kell, de az Hour of Code-ra készített bemutatkozó feladvány megoldásához ezt nem szükséges megtenni.



Az oldalra érve a „Start New Game” gombra kattintva megkapjuk a kérdést, hogy mely programnyelvet szeretnénk használni. A Python és a JavaScript közül van lehetőség választani. A két nyelv között a feladat megoldásakor minimális különbséget fogunk csak érezni, azt használjuk, amelyiket szeretnénk. A legszembetűnőbb különbség, hogy Python esetén a különböző sorok külön utasításokat jelölnek, a JavaScript esetén minden utasítás után pontosvesszőt kell írni.



Az első pályán a főhőst kell irányítanunk: anélkül, hogy tuskébe szaladna, el kell érje a drágakövet.



A főhősnek jobbra, balra, fel, illetve lefelé van lehetősége mozogni, a „Segítség” blokkban javasolt utasítások alapján. A segítség mint súgó megmondja, hogy mely utasítást használhatjuk, kódoláskor azonban további segítséget is nyújt a CodeCombat kódíró rendszere: automatikus kódkiegészítés funkcióval rendelkezik, azaz miközben gépeled az utasítást, felajánlja a lehetséges választ, és enter gomb lenyomására el is fogadhatod azt.

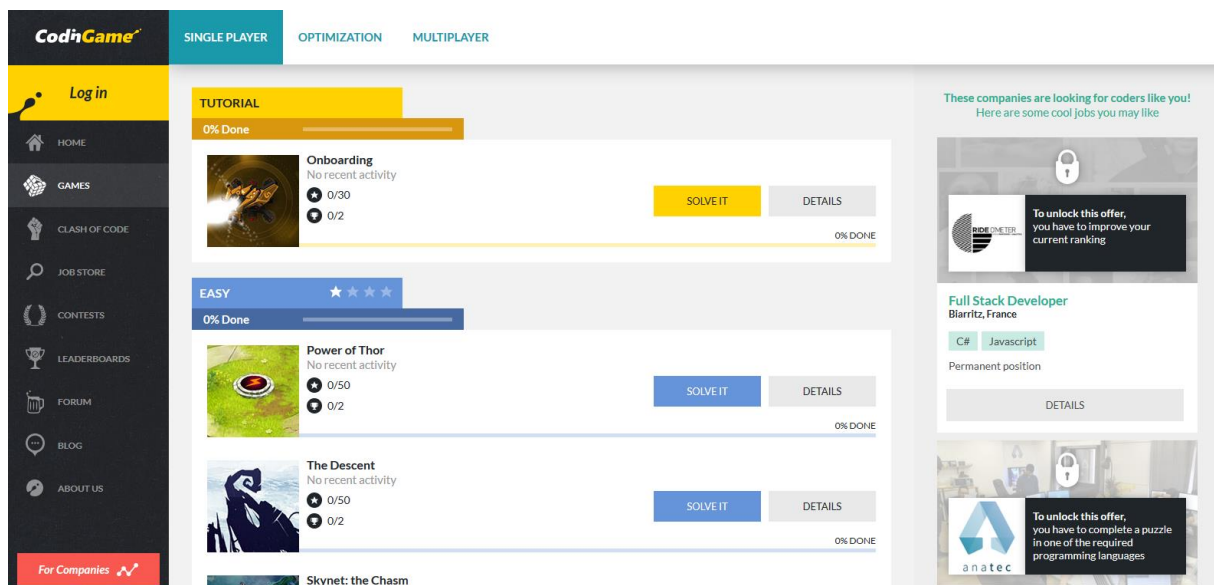


Fent, a „Játék menü” gombra kattintva további segítségek érhetőek el introvideó illetve rövid leírás formájában.

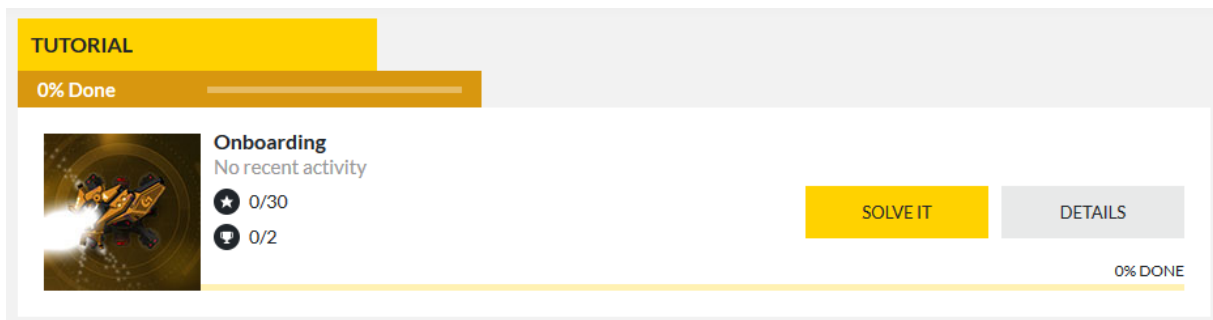


## 9. CodinGame

A CodinGame egy olyan angol nyelvű rendszer, amelyben több programozási nyelven lehetséges megoldást keresni összetett problémákra. Bár regisztrálást javasol több helyen, valójában a használatához ez nem szükséges.



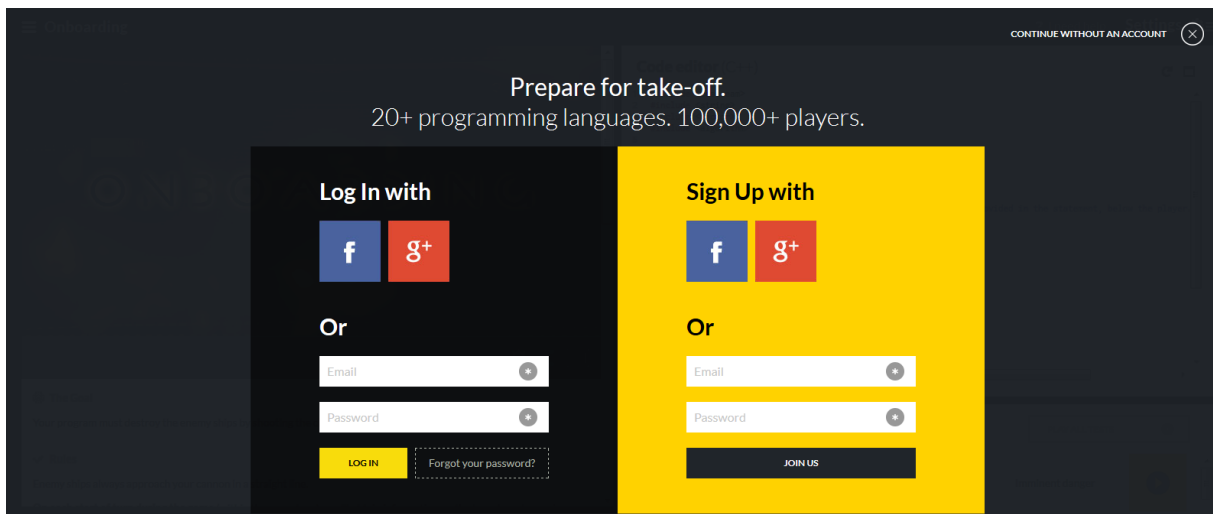
A weboldalt meglátogatva, a „Games” menüpont alatt a „Tutorial” problémával célszerű kezdeni, amely kezdő szintű ismereteken keresztül a rendszer működését is bemutatja egy egyszerű problémával.



The screenshot shows a user interface for a tutorial. At the top, there is a yellow header with the word "TUTORIAL" and a progress bar below it indicating "0% Done". Below the header, there is a section for "Onboarding" with a small image of a rocket launch. To the right of the image, it says "Onboarding" and "No recent activity". Below that, there are two circular icons: a star icon with "0/30" and a shield icon with "0/2". To the right of these icons are two buttons: a yellow "SOLVE IT" button and a grey "DETAILS" button. At the bottom right of the section, there is a small "0% DONE" indicator.

A „Tutorial” fül alatt szereplő, „Onboarding” című játéknál a „Solve It” gombra kell kattintsunk.

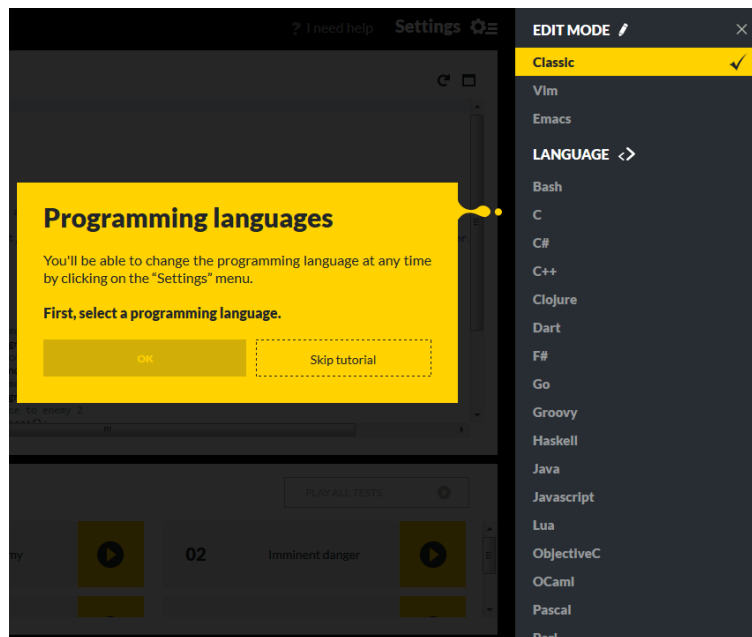
Ekkor a keret elkezdi betöltődni, majd felugrik egy ablak, amely bejelentkezésre vagy regisztrációra kér. A jobb felső sarokban található „Continue without an account” lehetőséget választva továbbléphetünk, ekkor a haladásunk nem kerül mentésre.



The screenshot shows a dark-themed login and sign-up screen. At the top right, there is a link "CONTINUE WITHOUT AN ACCOUNT" with a close icon. The main heading is "Prepare for take-off." followed by the text "20+ programming languages. 100,000+ players." Below this, there are two main sections: "Log In with" on the left and "Sign Up with" on the right. Both sections have a yellow background. The "Log In with" section has buttons for Facebook (f) and Google+ (g+), followed by "Or" and input fields for "Email" and "Password". There is a "LOG IN" button and a "Forgot your password?" link. The "Sign Up with" section has buttons for Facebook (f) and Google+ (g+), followed by "Or" and input fields for "Email" and "Password". There is a "JOIN US" button.

Ezután a keret rögtön arra kér, hogy válasszunk szerkesztési módot, és programozási nyelvet.

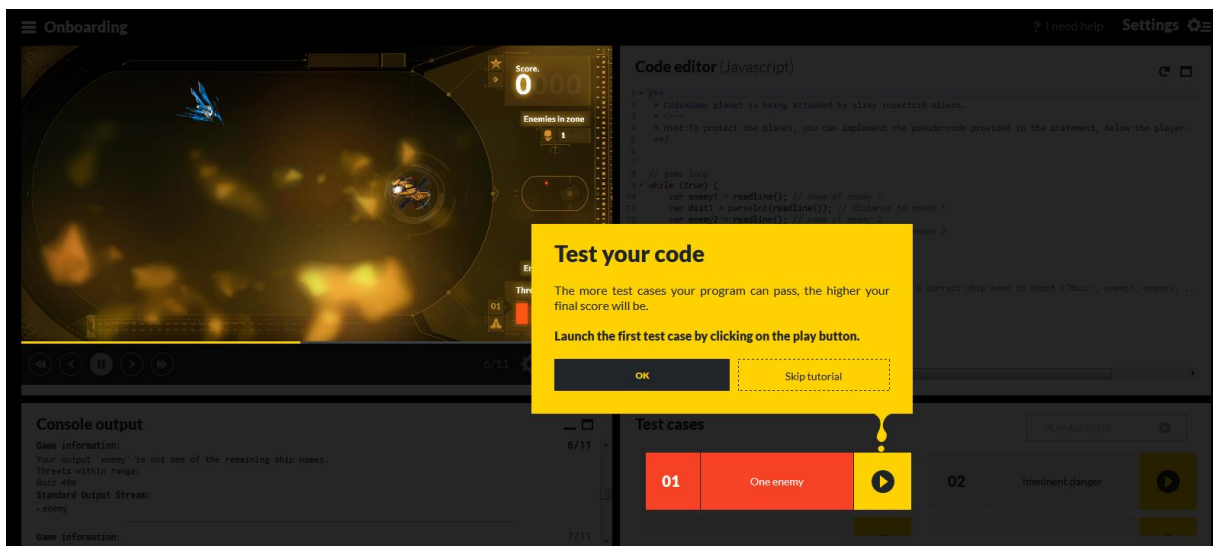




A hosszú listából a klasszikus (alapértelmezett) szerkesztés a javasolt, a programozási nyelv pedig ízlés kérdése: ha nem használtad még egyik felsoroltat sem, akkor a JavaScript vagy a Python kiválasztása a javasolt. A feladat ugyanúgy kényelmesen megoldható C#, C++ vagy Java nyelven is.

A képernyő bal felső részében látható a játék vizuális ábrázolása, alatta a feladat leírása, és a megoldás javasolt lépései, ötletek. A jobb oldalon látható a munkaterület, a kódszerkesztő, és alatta a megoldandó tesztesetek.

Ha kiválasztunk egy tesztesetet, a programunk futtatása elindul, és a viselkedés a jobb oldalon látható, alatta konzolos kimenetként olvasható. Ha szükséges, a hibakonzolra is van lehetőségünk írni: az utasítás nyelvenként más-más, pl. JavaScriptben `printErr()`.



A tutorial bemutatja azt, hogy hogyan lehet a felületet használni, majd magunkra hagy: kezdhetjük a kódolást!

A továbbiakban az egyes feladatok megoldásának segédlete található. javasolt először önállóan elkezdni, és csak ha elakadtál, akkor használd! :)

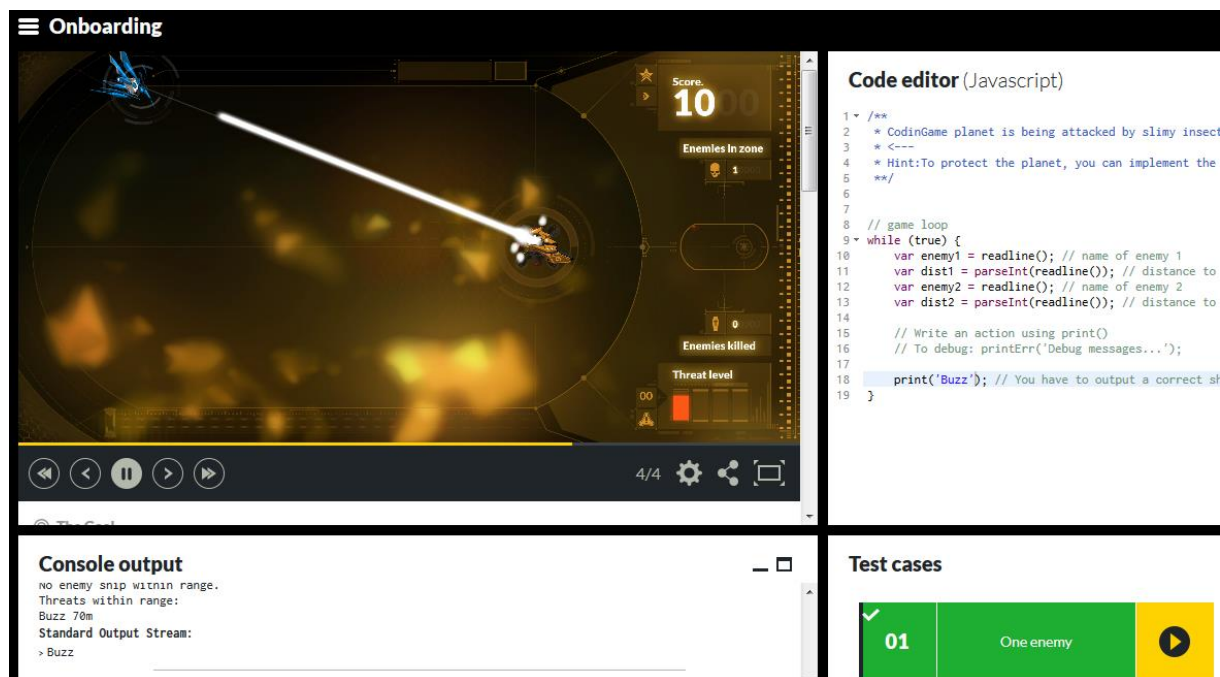
## a. Onboarding

Az Onboarding című játékban egy bolygó védelméért felelünk: a bolygót támadó ellenséges űrhajók lelövésére kell algoritmust készítenünk.

A játék egy végtelen ciklusban játszódik: minden kör a ciklus egy fordulója. A kör elején adatokat kapunk a közeledő ellenfelek nevééről, és a kör kezdetén mérhető távolságáról. Ezeket konzolról való beolvasás reprezentálja – de természetesen nem szükséges semmit bevinni, a környezet megteszi helyettünk.

Amit meg kell oldani, az a konzolra való kiíratás olyan jellegű megvalósítása, hogy egy ellenfél neve kerüljön kiíratásra, amelyre az általunk irányított ágyú löni fog. Ha rossz nevet adunk meg, nem történik lövés, ha az ellenfél túl közel kerül, akkor az ágyú felrobban.

Például, ha a JavaScriptes forrást választjuk, akkor a kód végén a konzolra kiírató, print() utasításnak a „Buzz” paramétert adva, az első tesztet teljesíthető.



The screenshot displays the 'Onboarding' game interface. On the left, a space battle scene shows a blue spaceship firing a white beam at a yellow enemy ship. The right side features a 'Code editor (Javascript)' with the following code:

```
1 /**
2  * CodinGame planet is being attacked by slimy insect
3  * <---
4  * Hint:To protect the planet, you can implement the
5  */
6
7
8 // game loop
9 while (true) {
10   var enemy1 = readline(); // name of enemy 1
11   var dist1 = parseInt(readline()); // distance to
12   var enemy2 = readline(); // name of enemy 2
13   var dist2 = parseInt(readline()); // distance to
14
15   // Write an action using print()
16   // To debug: printErr('Debug messages...');
17
18   print('Buzz'); // You have to output a correct st
19 }
```

Below the code editor is the 'Console output' window, which shows the following text:

```
no enemy snip within range.
Threats within range:
Buzz 70m
Standard Output Stream:
> Buzz
```

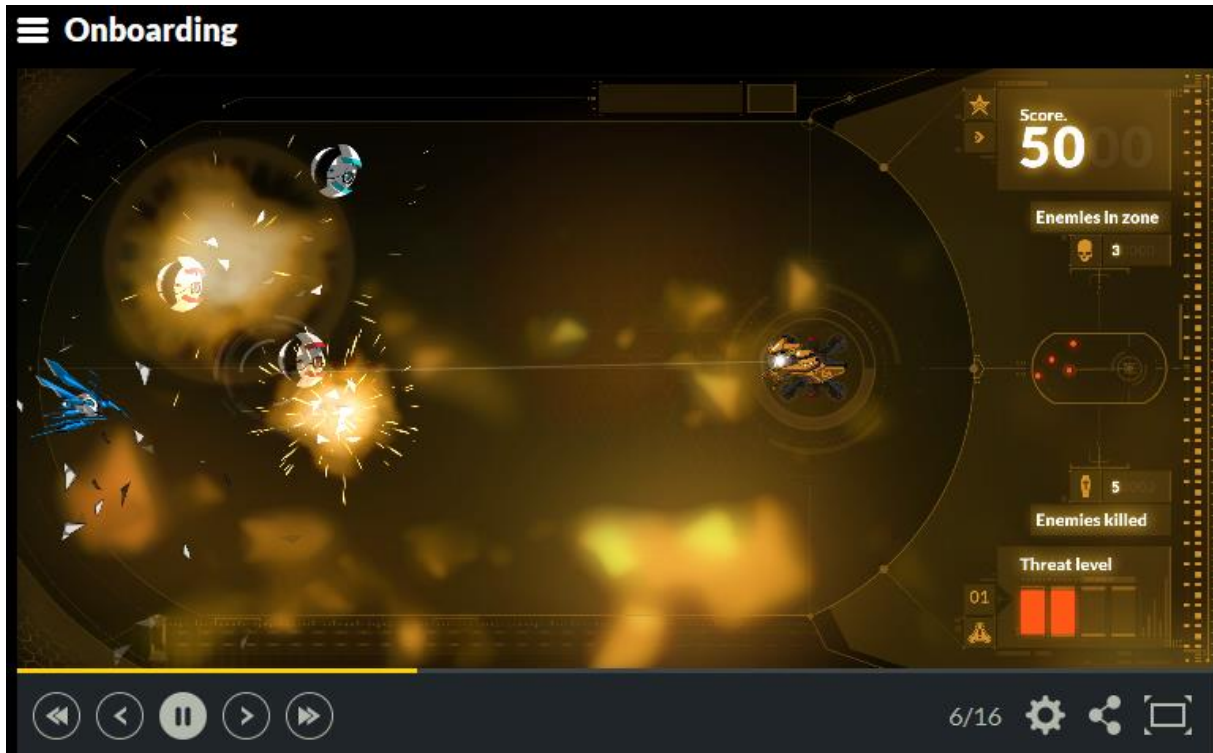
At the bottom right, there is a 'Test cases' section with a green bar labeled '01' and 'One enemy', and a play button icon.

Sajnos a második, harmadik vagy negyedik tesztet kipróbálva, láthatjuk, hogy ez nem általános megoldás, hiszen ha sokkal több beérkező ellenfél van, és a nevek is váltakoznak, előbb utóbb mindig legyűrnek.

Javasolt tehát a kapott változók közül kiírni azt, amelyik közelebb van, hiszen a távolabbat vélhetően lesz még idő lelőnünk.

Mondatszerű leírásban:

HA  $dist1 < dist2$  AKKOR  
KI: enemy1  
KÜLÖNBEN  
KI: enemy2

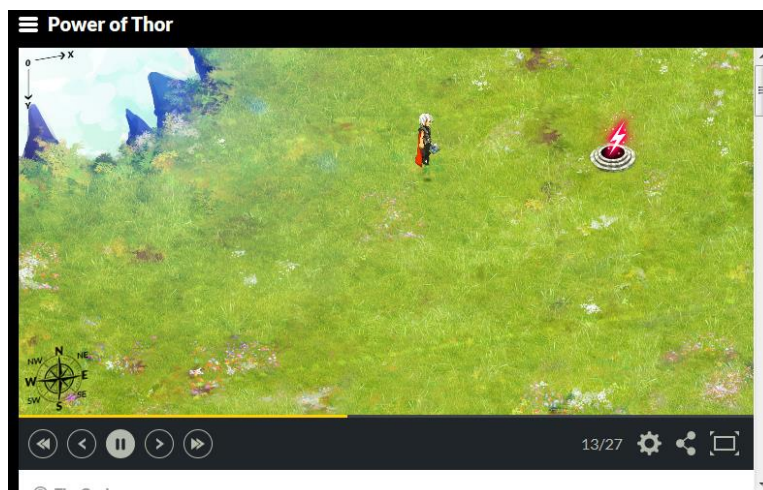


Helyes megoldás esetén az összes teszt teljesül.

## b. Power of Thor

Ebben a játékban Thornak kell segíteni a fényhez eljutnia. A játék kezdetén ismert Thor, valamint a célfény kezdőpozíciója a térképen (X és Y koordináták, mindkettőnél), melyet virtuális beolvasással kapunk, csakúgy, mint a bevezető feladatnál.

A kimenet az iránytű egy iránya kell legyen, betűkkel megadva: N NE E SE S SW W vagy NW.



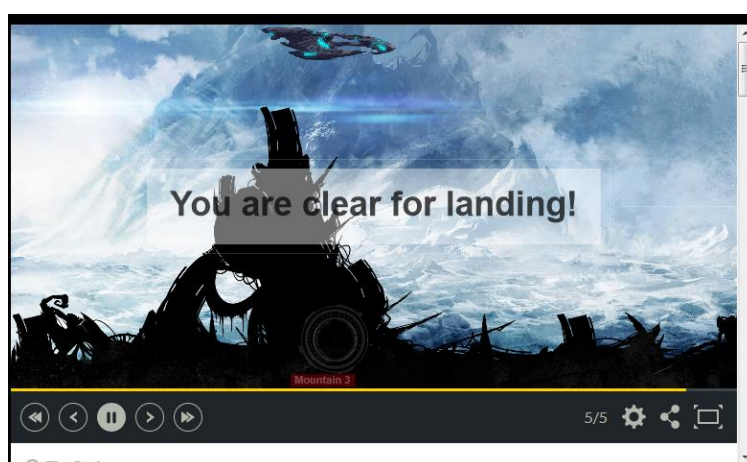
Egy olyan megoldást célszerű készíteni, ahol a játékos aktuális helyzetét is követjük a kiadott utasítások nyomán, és az új lépéseket is ebből számítjuk ki. Ez a főirányokban (N, S, W, E) való mozgásnál gyors megoldás, átlósan azonban nem feltétlenül a legjobb megoldás.

Tipp: feltételes szerkezetek egymásba ágyazásával átlós mozgást is definiálhatunk!

### c. The Descent

A játék lényege, hogy egy csillaghajó bombáit kell időzíteni, mégpedig úgy, hogy pontosan akkor tüzeldjen, amikor egy olyan hegy fölé ér, amibe folyamatos süllyedése végett előbb utóbb beleszapódna.

A feladat: megtalálni a legmagasabb hegyet, és pozícióját, és amikor a hajó pontosan felette van, tüzeldjen, egyébként „HOLD” utasítással ne tegyen semmit.



A kör elején megkapjuk a hajónk helyzetét, X és Y koordinátákkal. Az X koordináta gyakorlatilag azt jelöli, hogy *hányadik* hegy felett vagyunk. Ezután 8 lehetséges hegymagasságot kapunk meg, balról jobbra haladva. Érdeemes megfigyelni, hogy a hegymagasságok cikluson belüli helyi változók, így kívül nem elérhetőek.

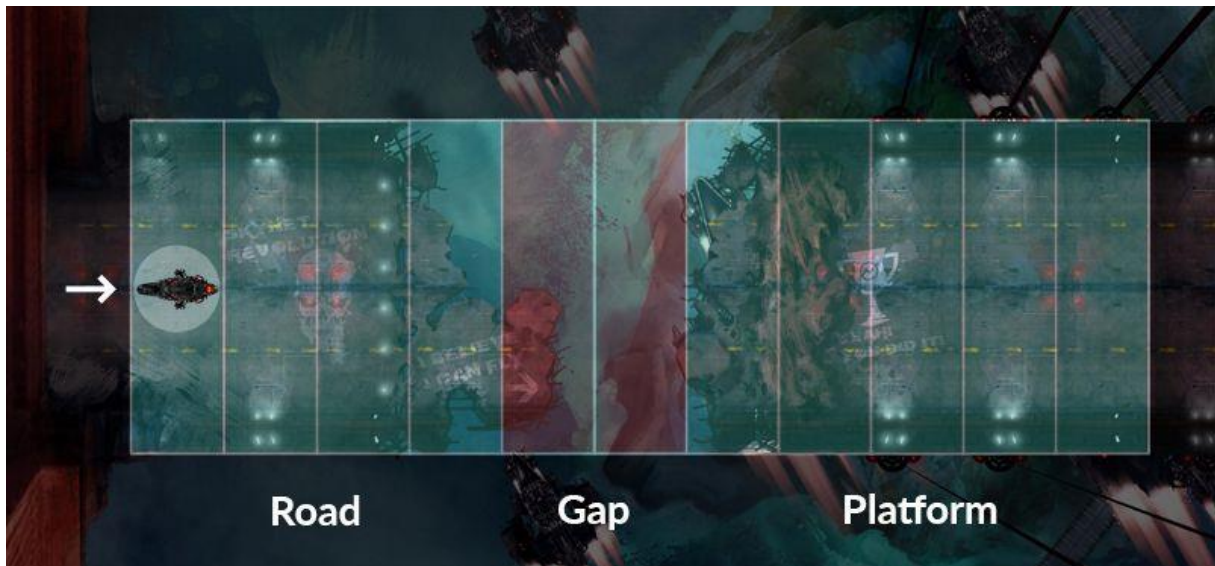
A megoldás javasolt menete: a ciklusból kiválasztani a legmagasabb hegy helyét, és ha felette vagyunk tüzelni, ellenkező esetben várakozás, amíg fölé érünk.

Tipp: ügyelj arra, hogy a cikluson vagy feltételen belül létrehozott változók a szerkezeten kívül nem elérhetőek!

#### d. Skynet: The Chasm

2027-ben járunk, miután a Skynet uralja a világot, egy meghackelt motoros terminátort kell irányítanunk. A feladata az, hogy egy félig lerombolt hídon felgyorsítson, átugorja a szakadékot, majd biztonságosan megálljon.

Négyféle utasítást tud végrehajtani a motoros: SPEED esetén gyorsul, SLOW esetén lassul egy egységgel, JUMP utasításra ugrik, WAIT esetén az adott körben nem tesz semmit.

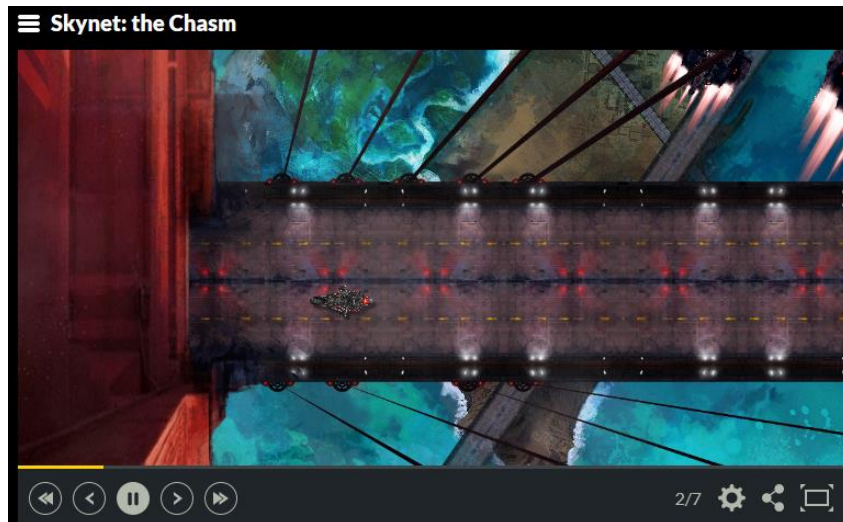


A játék kezdetén megkapjuk az előttünk álló út, a szakadék, és a szakadék utáni platform hosszát. Úgy kell számolnunk, hogy amikor elfogy az út, akkor ugrasson a motoros, majd a túloldalon való földet érés után megállni is tudjon.

Minden kör elején megkapjuk a motoros sebességét, illetve az aktuális helyzetét, és a kör végéig ki kell adnunk egy utasítást.

Hajrá!





Tippek: a rés átugrásához szükséges sebesség  $gap+1$  értékű. Ha nagyobb sebességgel ugrik a motoros, távolabb ér földet, és nehezebb lehet lelassulnia a platformon úgy, hogy ne essen le. Gyorsuljon fel a motoros mielőbb, és ugorjon az utolsó lehetőségénél, amikor még nem zuhan le, majd földet érve fékezzen, amíg meg nem áll!