

I. 348.

Kovács Balázs Marcell

Budapest, ELTE Radnóti Miklós Gyakorló Iskola, 11. o. t.

kbalazsm@gmail.com

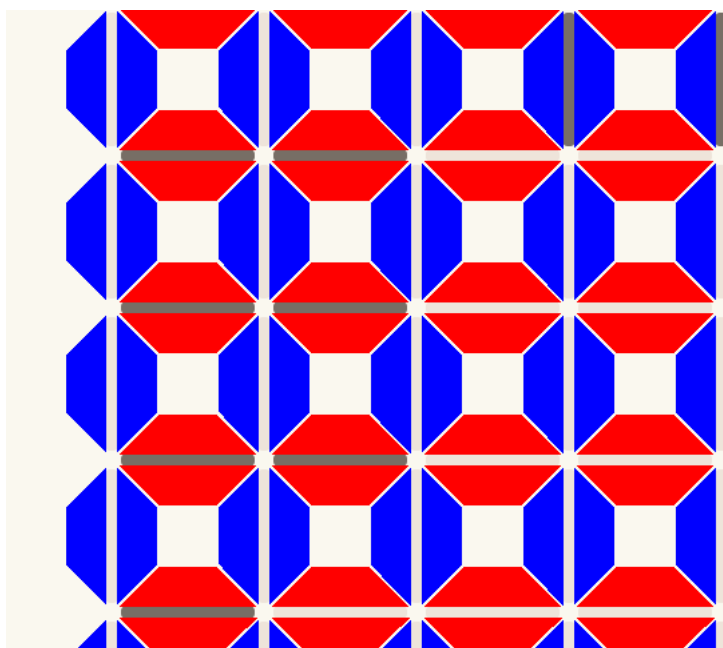
A megoldás JavaScript használatával készült, így fordítani nem szükséges. A játék egy HTML-es oldalon jelenik meg, ennek ellenére természetesen offline játszható.

A játék kizárólag Mozilla Firefox 29 böngészőben működik tökéletesen (valószínűleg néhány korábbi FF verzió is megfelelő). Utólag törekedtem arra, hogy Google Chrome-ban is működjön (34-es verzióban teszteltem), működik is, azonban néhány funkció egyáltalán nem, vagy hibásan, akadozva (animációk) működik.

A JS kód 4 részre van bontva, megvalósul a logika, a bemenet és a megjelenítés (majdnem) teljes szétválasztása. A kód struktúráját a 2048 nevű szintén HTML-es játék forrása alapján mintáztam. Kényelmi okokból a jQuery könyvtárat használtam a megoldáshoz.

A lap stílusának leírását a **style/sass** mappában található file-ok tartalmazzák (ezek elkészítéséhez kellett a legtöbb idő). Ezek **scss** file-ok, ezeket kell csak ténylegesen “fordítani”, ehhez szükséges a **sass** nevű eszköz. Utólag a különleges CSS3 szabályokat megpróbáltam általánosra alakítani a Bourbon könyvtár beépített makrói segítségével, hogy több böngészőben is helyesen jelenjen meg a lap. A játék több mindenben erősen hagyatkozik CSS3, illetve HTML5 elemekre, ezért van mindenképp szükség modern böngészőre.

Maga a játék egy HTML-es táblázat, aminek cellái a kis négyzetek egyes oldalai. Ezekre kattintva lehet megtenni a lépéseket. Az oldalak viszonylag keskenyek, így nehéz őket egérrel eltalálni, pláne ha sietni kell az 5s-es időlimit miatt, ezért az oldalakhoz közel nagyobb terület is érzékeny a kattintásra, ezt a következő ábra szemlélteti. Természetesen a játékban ezek a trapézok átlátszóak, ahhoz hogy megjelenjenek, a **jatek.scss** elején kell a **\$hatarok** változót **true**-ra állítani (majd lefordítani css-re). Néhány design-elem (háttér és néhány egyéb szín) szintén a 2048 játékból lett másolva.



## A játék funkciói

A böngészőablak átméretezésekor az egyes elemeket úgy igazítjuk, hogy maga a játéktérület a lehető legkevésbé legyen összenyomva (az eredeti méreténél soha nem lesz nagyobb). Ez főleg kis képernyőknél előnyös. Ugyanakkor egy kattintással teljes képernyős módba válthatunk, ezzel megnövelve a rendelkezésre álló területet. Egy ilyen váltásnál a játék automatikusan megáll. Túl kicsi képernyőméretnél a leírás összehúzódik és görgethetővé válik.

Az animációkat ki-be lehet kapcsolni: ha nem elég gyors a gép, akkor az akadózó animációk zavarók lehetnek (Intel® Core™ i3-2120 CPU @ 3.30GHz  $\times$  4 processzorral, ill. Intel® Sandybridge Desktop grafikával tökéletesen működnek az animációk).

Lehet állítani, hogy egy lépés megtételéhez teljes kattintás (lenyomás + felemelés) vagy csak gomblenyomás szükséges. Ezért azért fontos, mert ha a felhasználó kattintás közben mozgatja az egeret (ez igen valószínű) és először rákattint egy trapézra, majd továbbhúzza az egeret és a másik trapézon emeli fel, akkor ezt a program nem érzékeli kattintásnak, mert a felemelés már egy másik elemen történt.

Csúszkákon lehet állítani a pálya méretét ( $n$ ) és az előre elhelyezett oldalak számát ( $k$ ). A csúszkák mozgatásakor a program automatikusan rajzolja az új méretű pályát, illetve elhelyez megfelelő számú oldalt (sorban).

Új játék kezdése előtt egy 5s-es visszaszámlálás van.

A játékot meg lehet állítani a *Szünet* gomb, a *space*, vagy a *P* billentyűk megnyomásával. Folytatni a *Folytatás* gombbal, a *space* vagy a *C* billentyűvel lehet. Amíg a játék szüneteltetve van, a játék állása nem látszódik.

A képernyő jobb szélén található a pont- és az időszámláló, valamint egy nagy méretű visszaszámláló (5s-ből). Ez alatt egy csík is található: ha elfogy a csík, lejárt az idő és jön a játék vége üzenet.

Hibás lépés után az elzárt kisebbik terület pirosra színeződik és megjelenik a játék vége üzenet.

## A játék logikája

Egy tömbben tároljuk, hogy melyik oldal van már felhasználva, és melyik nincs. Egy másik tömbben tárolunk minden egyes mezőhöz (négyzet) egy listát az elérhető szomszédos mezőkről.

Kezdetben az oldalakat véletlenszerűen helyezzük el: a vízszintes oldalak elhelyezésénél az egyetlen kritérium, hogy ne legyen olyan sor, ami nem a nagy négyzet szélén van és tele van behúzott oldalakkal. A függőleges élek behúzásakor pedig mindig megvizsgáljuk, hogy az összefüggőség teljesül-e. Az összefüggőség vizsgálata egyszerű szélességi kereséssel történik: a behúzott új él egyik oldaláról akarunk eljutni a másikra. Ha ez lehetséges, akkor a terület összefüggő (feltételezve, hogy eddig is az volt).

## További funkciók

Bár a játék kész van és bőven megfelel a szabott feltételeknek, még mindig van rajta mit fejleszteni. Ide tartozik például egy *Are you sure* típusú dialog játék megszakítása előtt, valamint rekordok kezelése. Ez utóbbi problémás, ugyanis a pálya méretétől és az előre elhelyezett oldalak számától jelentősen függ az elérhető maximális pontszám (tervezek egy algoritmust is, ami megmondja egy adott véletlen pályáról az elérhető maximális pontszámot).