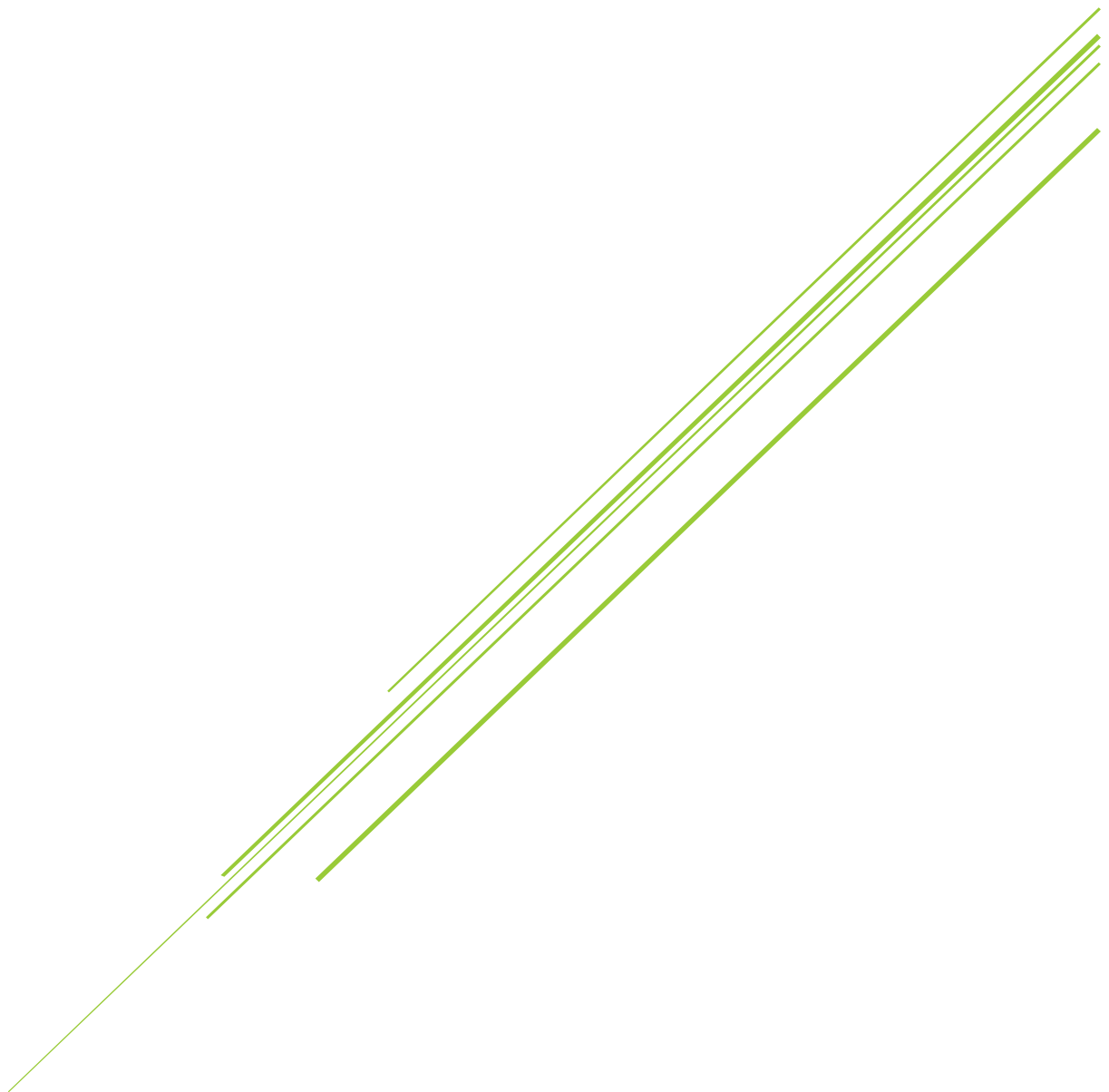


Töltés játék

Tesztelési dokumentáció



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Programozás alapjai I.

Tartalom, a tesztelés terve

TÖLTÉSEK.....	2
TÖLTÉSEK TÖRLÉSE, A LÁNCOLT LISTA TESZTELÉSE	2
FIX TÖLTÉSEK MÓDOSÍTÁSA	2
HÁROMNÁL NAGYOBB ABSZOLÚT ÉRTÉKŰ FIX TÖLTÉSEK.....	2
FALAK	2
PROTON FALLAL VALÓ ÜTKÖZÉSE	2
PROTON TÖBB FALLAL VALÓ ÜTKÖZÉSE (SAROK).....	2
TÖLTÉSEK FALBA LERAKÁSA.....	2
RÁCSPONTOK ÉS MEMÓRIA.....	3
DICSŐSÉGLISTA	3
HIÁNYOS FÁJL BEOLVASÁSA	3
JÓ FÁJL BEOLVASÁSA	3
PÁLYÁK	4
HIBÁS PÁLYÁKAT TARTALMAZÓ FÁJL OLVASÁSA	4
HELYES PÁLYÁKAT TARTALMAZÓ FÁJL OLVASÁSA	4
MENÜ	4
STRESSZ TESZTELÉS	4
SZÍNEK	4
SZÍNEK VÁLTÁSA KÜLÖNBÖZŐ KÉPERNYŐKÖN	4

Töltések

Töltések törlése, a láncolt lista tesztelése

A játékban a töltések egy láncolt listában vannak, így a játéknak nagyon fontos része a lista stabil működése. A fejlesztés során sokszor fordultak elő „véletlenszerű” összeomlások a lista kezelésében való hibák miatt. A kész program tesztelésének eredményei:

A játékos által lerakott töltés törlése: három lerakott töltés közül az első, a középső, és az utolsó is gond nélkül törölhető. A három töltést a protonhoz közel lerakva látjuk, hogy hatásuk is ténylegesen megszűnik, a proton mozgását nem befolyásolják a kitörlésük után. A törlések helyes működését az stdout.txt fájlban lévő log is megerősíti, itt nyomon követhető a lista pointereinek helyes átállítása.

Fix töltések módosítása

A fix töltések tesztelésében is a log fájl adta az alapot, mivel a helyes működés csupán az, hogy a kattintások hatására semmi sem történik a képernyőn. Itt látható volt, hogy a játék regisztrálta a kattintás az adott pontban, és meghívta a `Toltes_novel` függvényt, a töltések listája pedig az elvárásokkal megegyezően nem változott meg.

Háromnál nagyobb abszolút értékű fix töltések

Külön tesztet kaptak a háromnál nagyobb abszolút értékű töltéssel rendelkező töltések, mivel a játék először csak a maximum három abszolút értékű töltések kezelésére lett tervezve (mivel ezek a felhasználó által létrehozható legnagyobb töltések).

Tapasztalatok: a háromnál nagyobb számok kiírásra kerülnek a töltések fölött, habár a tíznél nagyobbak nincsenek pontosan középen a töltés felett. Ez elfogadható, mivel valószínűleg soha sem lesz szükség ekkora töltésre, a proton és közte fellépő erő úgyis játszhatatlanná tenné a pályát.

A tesztelés megerősíti, hogy már a -10 töltésű töltés is azonnal magába rántja a protont. Ebből az is látható, hogy a nagyobb töltések ténylegesen nagyobb erővel hatnak a protonra, ami szintén a helyes működés.

Falak

Proton fallal való ütközése

A fallal ütközés már a fejlesztés alatt is sokszor tesztelve lett, a fal pirossal rajzolásának optimalizálása miatt. A kész játékban is az elvárt módon működik a fal, ha a proton sebessége a fal felé mutat, akkor nullára változik. Nem ragad azonban a falhoz, ha például a fal másik oldalára elhelyezünk egy pozitív töltést, az el tudja lökni ismét a faltól a protont.

Proton több fallal való ütközése (sarok)

A proton helyesen kezeli a több fallal való ütközést is, és „beragad” a sarokba, viszont kihozható onnan, ha megállítjuk a játékot, és a sarok túloldalára pozitív töltéseket helyezünk el.

Töltések falba lerakása

A teszteléshez egy (500, 200) koordinátájú rácspontba próbáltunk töltést elhelyezni. Emellett egy függőleges fal helyezkedett el, ennek x koordinátáját változtattuk 1-es léptékkel. A töltést mindaddig le lehetett tenni, amíg a fal koordinátája 510-nél nagyobb, vagy 490-nél kisebb volt. Ez a várt eredmény, mivel a töltések sugara alapértelmezetten 10.

Eközben a teszt közben szintén teszteltük a pálya újratöltését, mivel kilépés nélkül, csak a pálya gombjának újbóli megnyomásával töltöttük újra a pályát a fal adatainak a pályákat tartalmazó fájlban való átírása után. Ez is kifogástalanul működött.

Ráncpontok és memória

A ráncpontok tömbjének kapcsán fejezzük be a memóriakezeléssel kapcsolatos teszteket. A játék memóriaszivárgásra az infoc.eet.bme.hu oldalon található `debugmalloc` függvénykönyvtárral lett tesztelve. A tesztelés közben sem normális leállásnál, sem hiba miatti leállásnál nem jelzett hibát a `debugmalloc`.

A hiba miatti leállás a toplista fájl átnevezésével, illetve a pályákat tartalmazó fájl elrontásával lett előidézve, ezekről bővebben később.

A biztonság kedvéért maga a `debugmalloc` könyvtár is tesztelve lett, a kód felszabadítást végző részének kommentelésével szándékos memóriahibát okozva tényleg jelezte a szivárgást.

Dicsőséglista

Hiányos fájl beolvasása

Normális esetben a dicsőséglista fájl elejét és végét is a

```
| 0;0
```

sor jelöli. A tesztelés kétféle hiányos fájlal foglalkozott. Ha hiányzott az első ilyen sor, vagy nem volt elég sor a dicsőséglistában (a `hsdb` változó értéke, alapértelmezetten 10), akkor a fájlban lévő rekordokat helyesen olvasta be a program, a maradék helyet pedig kitöltötte nullákat tartalmazó sorokkal. Ez elfogadhatóan néz ki a dicsőséglistán, láthatóan az üres helyeket jelöli, és idővel az új eredmények úgy is eltűntetik őket. A program futásának végén, a lista fájlba visszaírásánál már megjelenik az első (eredetileg hiányzó) sor, a program a hibának ezt a részét tehát kijavítja.

Ha az utolsó, lezáró sor hiányzott, a program a dicsőséglista fájl sikertelen megnyitását jelző hibakóddal kilépett. Habár ez nem pontos jelentése a hibának, a log fájl olvasásából kiderül a pontos probléma.

Jó fájl beolvasása

Egy helyesen formázott fájl beolvasásakor jól működött a beolvasás, az új rekord felvétele, az egész lista sorba rendezése, és fájlba vissza írása. A teszteléshez használt fájl:

```
| 0;0  
| 2500;ALFA  
| 12000;BETA  
| 3250;CHARLIE  
| 7250;DELTA  
| 9250;ECHO  
| 19750;FOXTROT  
| 10750;GOLF  
| 11500;HOTEL  
| 14000;INDIA  
| 11500;JULIETT  
| 0;0
```

Pályák

Hibás pályákat tartalmazó fájl olvasása

Tesztelt esetek:

- Az első, a játéklap méreteit megadó sor helytelen formátumban van megadva, vagy nincs megadva
- A pálya öt alap adatából hiányzik valamelyik
- A pálya falakat tartalmaz, de hiányzik az F karaktert tartalmazó sor
- A pálya töltéseket tartalmaz, de hiányzik a T karaktert tartalmazó sor
- Hiányzik a pályát lezáró E karaktert tartalmazó sor

Az eredmény az összes fenti esetben azonos: a játék bezáródik, megjelenik a pálya sikertelen beolvasásakor várt hibakód.

Ezekon kívül tesztelve lett az az eset, amikor szerepel egy F vagy T karaktert tartalmazó sor, de azok után nincsenek falak és töltések adatai megadva. Ilyenkor a logban megjelenik, hogy falak/töltések vannak az adott pályán, de egy sem lesz beolvasva, a játék hiba nélkül fut.

Ha a fenti esetek nem az első pályán voltak, akkor az adott pálya betöltésekor lépett ki a játék (a konkrét teszt a második pálya hibás megadásával történt).

Helyes pályákat tartalmazó fájl olvasása

A helyesen formázott pályák rendben működtek, a pályák paraméterei jól töltődtek be.

Menü

Stressz tesztelés

A menü egyetlen külön tesztelése a véletlenszerű kattintgatás és gyorsgomb nyomkodás volt, aminek célja a menü valamilyen elrontása, kiakasztása volt. Ezek közben a menü megfelelően működött, csupán néha lelassult a válaszideje az ismételt fájlolvasási műveletektől az új pályák betöltésekor.

Színek

Színek váltása különböző képernyőkön

A képernyő rajzolását három függvény végzi, így három tesztre van szükség.

A játék futása közben (a proton mozgásától függetlenül), a pálya végén megjelenő képernyőn, és a dicsőséglista nézetben is mindig működött a színek váltása.

Ezen kívül a random színek generálása is tesztelésre került, a random színséma kiválasztásakor mindig új random színeket kaptunk, így ez a funkció is működőképes (habár hasznossága megkérdőjelezhető).