Törő Anna Nina (GH307A)

Aknakereső dokumentáció

Az aknakereső játékban a játékos -vagy felhasználó- egy általa meghatározott méretű pályán (és általa meghatározott bombaszámmal) játszik, azaz tippelget egy-egy mezőre -vagy blokkra-, hogy az nem bomba-e. A játék célja, hogy minden bombát megtaláljon, vagyis bezászlózzon úgy, hogy egyiket se nyitja fel. A nem bombás mezők ebben segítségére vannak, hisz mindegyik egy számot rejt, ami megmondja, hogy a mező 8 tagú szomszédságában hány bomba található.

Felhasználói dokumentáció:

A program megnyitásakor a felhasználó egy menürendszerben találja magát, ahol két opció közül választhat. Indíthat új játékot vagy megnézheti a toplistát.
Ha új játékot indít, kiválaszthatja a játék nehézségét, ami lehet könnyű, közepes és nehéz, vagy akár egyedi játékot is indíthat. Ebben az esetben egy pályatervezőben találja magát a felhasználó. A pályatervező két dolgot tesz lehetővé: megadható hány blokk széles és hosszú legyen a pálya, és hány bomba helyezkedjen el rajta. A szélesség és hosszúság értéke mindig megegyezik.
Végül egy nevet is meg kell adnia, ami alapján be tudja majd azonosítani magát a toplistán. Ezt az előre beállított nehézségű játékindításoknál is meg kell tennie.
Ezután elindíthatja a játékot, ahol a fentebb leírt lehetőségekkel rendelkezik, mint mező zászlózása (jobb egérgomb) vagy felderítése (bal egérgomb).
Ezeken felül újra is indíthatja a játékot az alul, középen található smiley-ra kattintva. A smiley mellett jobb oldalt látszik a játékos játékideje, míg bal oldalt a még kioszthatós zászlók száma. Amennyiben az nullát mutat, a játékos nem tud több zászlót elhelyezni. Ha ekkor még nem nyert, akkor a már zászlóként megjelölt mezők egyike biztosan tévesen lett megjelölve, és arra megint kattintva leveheti róla a zászlót és elhelyezheti egy új helyen.
Ha a toplista opciót választja, megnézheti az eddigi legjobb eredményeket, ahol megjelennek a megadott játékos nevek, a pálya beállításai és a játék ideje.

Programozói dokumentáció:

Az osztályokról:

A kezelőfelület megalkotásához Java Swing elemeket, illetve alacsonyszintű grafikai rutinokat használtam. Az osztályok nevei igen beszédesek, így már csak azok alapján is könnyen érthető, miért felelnek. A menürendszer egy Menu JFrame-ből, illetve egy arra illeszkedő CreateGame JPanel-ből áll, ami az aktuális menübeállítások alapján jelenik meg. Így, ha a játékos a szabványos méretek egyike szerint játszik, a CreateGame felülete csak egy nevet kér be, míg egyedi játékmódban mind a bombák, mind a pálya számát is bekéri, mindkettő JCombobox-ként jelenik meg, míg a név, egy egyszerű szöveges mező.

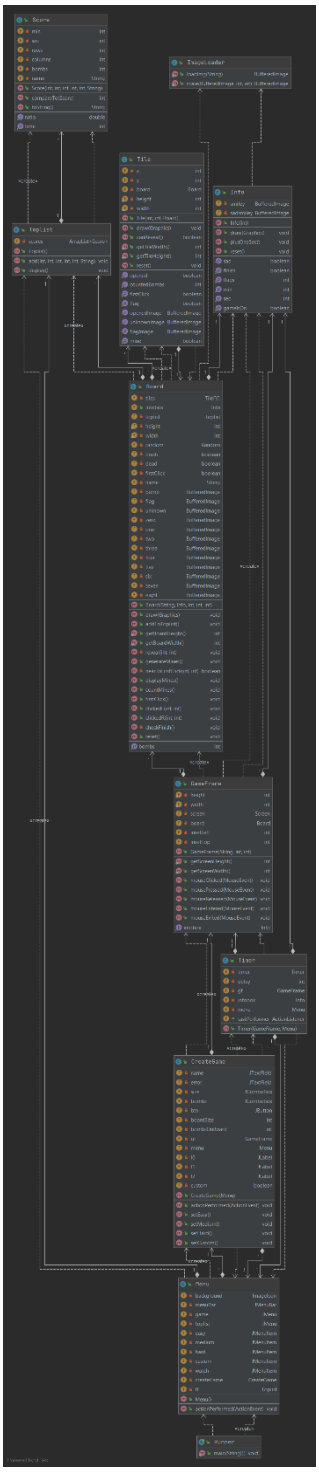
Ezután a program egy új JFrame-re vált, ez a GameFrame nevet kapta. Ezen helyezkedik el felül a Board osztály, ami a játékmezőt hivatott megvalósítani, illetve alatta az Info osztály. Az Info egyfajta infósávot jelöl, ahol játékhoz szorosan kötődő, de nem a játék részét képező információkat lát a felhasználó. Ide tartozik az újraindító gomb, az idő kijelzője, aminek a

megvalósításában egy Timer osztály van a segítségére, illetve a még kihelyezhető zászlók számát mutató doboz.

A Board osztály legfontosabb attribútuma a tiles[][] 2D-s tömb, ami a játékmezőn elhelyezkedő blokkokat foglalja egybe. Minden egyes blokkot a játékmezőn egy Tile osztály példány reprezentál.

A játék végén, ha nyert a játékos, egy új Score osztály példány generálódik, ami a győztes adatait tárolja, ez pedig hozzáadódik a Toplist osztály ArrayList-jéhez, ahol Score példányokat tárol, azaz a toplista elemeit.

Osztálydiagram:



A kép nagyon csúnyán megtörné a dokumentáció formázását, így ClassUML.png néven található meg a dokumentáció mellett. Ide csak egy kicsinyített változatát illesztettem be, hogy a dokumentáció minden lényeges eleme megtalálható legyen itt. Nagyobb, olvashatóbb formátumért külön képként mellékeltem.

Osztályok függvényeinek leírása:

Az osztályok függvényeinek, illetve attribútumainak leírása és részletezése közvetlen a forráskódban elérhető, JavaDoc formátumban.

Kimenetek és bemenetek:

A toplista elemit egy scores.dat nevezetű fájlban tárolom. A program indításakor ezt beolvasva jeleníti meg a menü a toplistát, amennyiben a Toplist -> Watch toplist útvonalat választja a felhasználó.

A játék végén, ha a felhasználó nyert, az aktuális adatokat kiírja a program ebbe a fájlba, a megfelelő helyre. Illetve elmenti az összes eddig belekerült adatot.

Use case-k:

New Game: A játékos létrehoz egy új játékot, aminek megadja nehézséget. Ezek a nehézségek a következők: könyű, közepes, nehéz és egyedi.

Flag square: A játékon belül a játékos megjelöl egy blokkot zászlózottként, tehát bejelöli, hogy ott bomba van, a blokk felnyitása nélkül. Ezt a jobb egérgombbal teheti meg.

Uncover square: A játékon belül a játékos megjelöl egy blokkot felnyitottként, tehát bejelöli, hogy ott biztosan nincs bomba. Ezt a bal egérgombbal teheti meg.

Game over: A játék ellenőrzi a játékos Flag square vagy Uncover square tevékenysége után, hogy a játék befejeződött-e. Ha igen, eldönti, hogy a játék vereséggel vagy győzelemmel ért véget.

Lose: A játék vereséggel ért véget.

Win: A játék győzelemmel ért véget, tehát felvihető az eredmény a toplistába.

Export Toplist: Ha győzelemmel ért véget a játék, az eredmény felkerül a toplistára. A toplista fájlja (scores.dat) szerkesztésre, majd mentésre kerül. A tartalma innentől az új eredményt is tartalmazza az eddigiek mellett.

Import Toplist: A játék megkezdése előtt, a játékos megtekintheti az eddigi toplistát, amihez a játék betölti a (scores.dat) fájlt és megjeleníti.