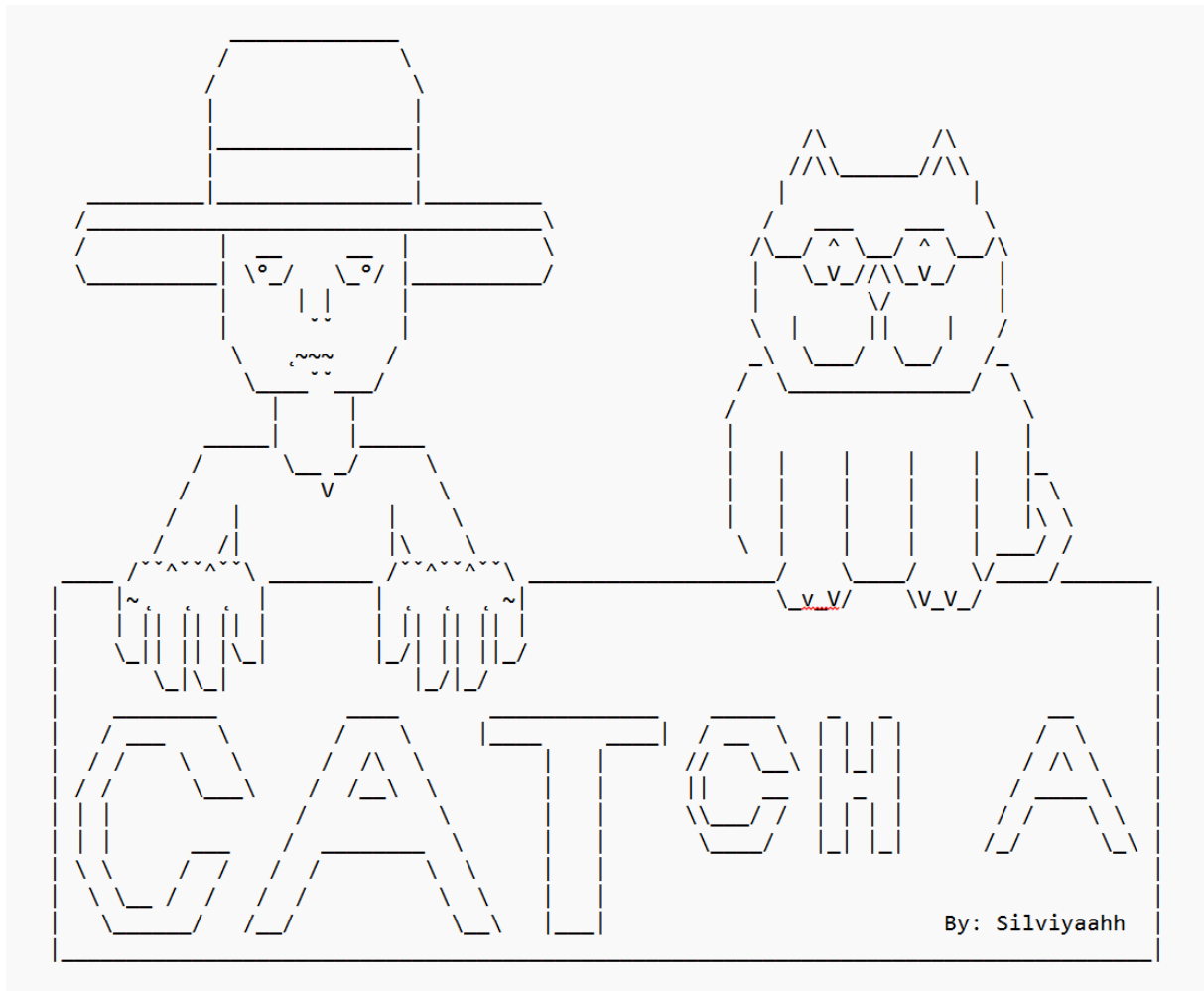


Programozás Alapjai 2

Nagy Házi Feladat

Specifikáció és Programterv



Az énáltalam választott házi feladat egy játék elkészítése.

Általános Információk

A játék neve: Catch The Cat (magyarra fordítva: Kerítsd be a Macskát).

A játék típusa: Multiplayer

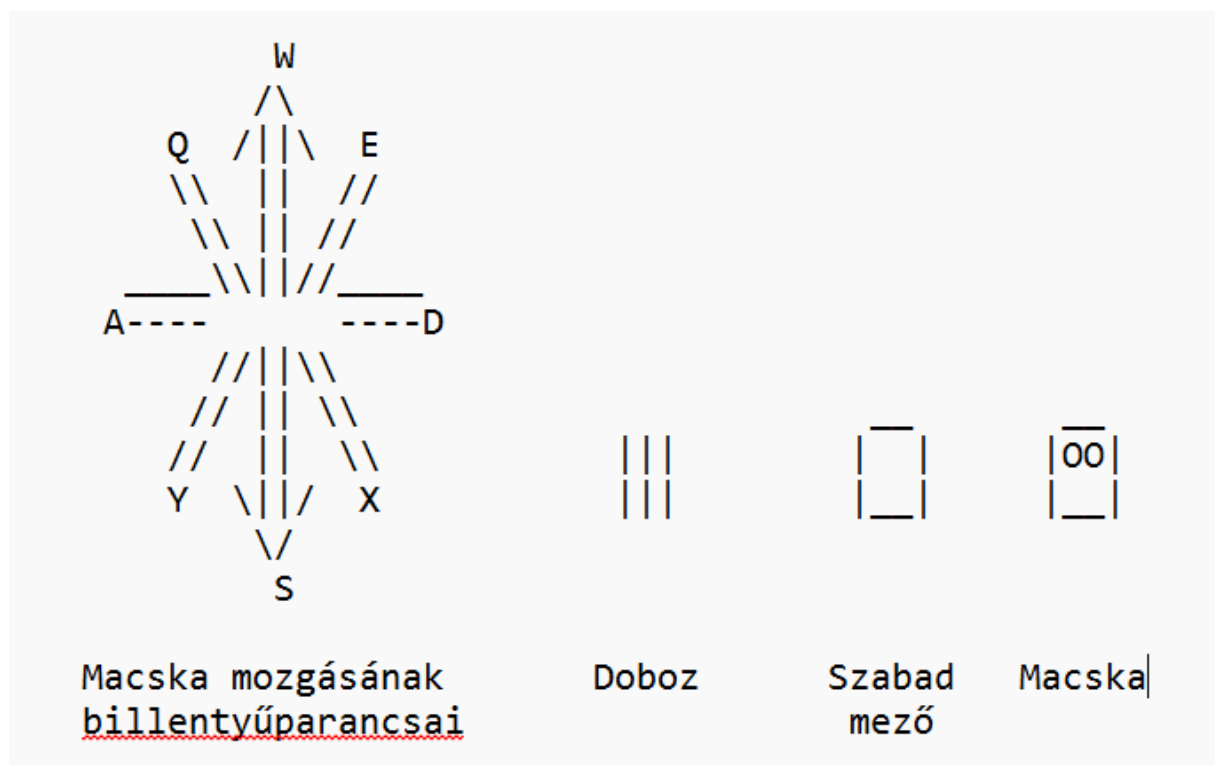
A játék célja: „Gazda” játékos részéről a macska bekerítése; „Macska” játékos részéről a Játéktábláról való kijutás.

A Játék menete: Először beregisztrál/bejelentkezik mindkét játékos, bejelentkezés után kiválasztják, hogy ki szeretne macska lenni és ki szeretne gazda lenni. Ezután létrehozzák a Játéktáblát.

A Játéktábla négyzet alakú mezőkből áll, tetszőleges alakú lehet (ezt a játékos tervezi meg). 3 féle mező létezik: a szabad mezők azok, ahova lehet követ rakni, illetve léphet rájuk a macska. Vannak viszont „kövek” vagy „dobozok”, amelyekre nem léphet a macska. Ezeknek a mezőknek a karaktergrafikus kijelzése a lenti ábrán látható. A játéktábla kezdetben is már tartalmaz néhány ilyen követ tetszőleges helyeken.

Ezután a gazda és a macska felváltva lépnek. A gazda a körében letesz egy követ annak érvényes koordinátáinak megadásával egy szabad mezőre. A következő körben a macska lép. Szabályok: a macska mindig az aktuális pozíciójának megfelelően lép a billentyűzet QWEASDYX gombjainak egyikével a tőle körben elhelyezkedő 8 mező egyikére, amennyiben azok szabad mezők, tehát nincs rajta doboz. A Macska nem tud átugorni egy vagy több mezőt, sem dobozt, és nem tud olyan mezőre lépni, ahol már doboz van.

A játéknak akkor van vége, ha a macska nem tud semerre mozogni (a körülötte lévő mind a 8 mezőn doboz van), vagy amennyiben a macskának sikerült elérnie a pálya bármely szélét. Az első esetben a gazda számára nyereséggel, a macska számára veszteséggel zárul a játék, míg ellenkező esetben értelemszerűen fordítva.



A játék további funkciói: Mentés során a játéktábla állapota, a játékosok kivolta valamint az is mentve lesz, hogy melyik játékosé a következő lépés. Mindezt egy txt dokumentumba menti. Ennek megfelelően természetesen be is lehet tölteni egy korábban elmentett játékot. Amennyiben egy mentett játékot beolvasunk, törlődik a mentettek közül. Lehetőség van továbbá véget ért játék táblájának állapotának mentésére is, ez egy külön txt fájlban történik.

Mindemellett értelemszerűen a felhasználók becenév-jelszó párosát is menti egy külön fájlba e-mail címükkel és mentőkérdésükkel/válaszikkal együtt, valamint, hogy hány játékot játszott az adott felhasználó összesen és abból mennyit nyert meg. Ugyanitt lehetőség van a profilod módosítására: felhasználónév vagy jelszó megváltoztatására, valamint elfelejtett jelszó javítására mentőkérdéssel vagy e-mailes azonosítással.

Újabb funkció lesz a játék értékelése: szöveges véleményt írhat a játékos, pontozhatja 1-től 5 csillagig a játékot, valamint lehetőség van mások kommentjeinek, értékeléseinek megtekintésére is. A pontozásokból elért átlag értékelést (az összes értékelések számának függvényében) megjelenítjük a kezdőképernyőben.

A játék teljes egészében karaktergrafikus megjelenítéssel fog működni.

Felhasználói Interakciók

A játékba belépve először ki kell választani, hogy magyar vagy angol nyelven szeretnék használni a programot. Mindkét esetben ugyanazok a műveletek végezhetők el, a szöveg nyelve az egyetlen változás, így a magyar nyelvűt fogom részletezni.

Ezután egy menüt látunk magunk előtt, melynek elemei:

- Új játék
- Mentett játék betöltése
- Mentett végeredmény kirajzolása
- Vélemények
- Értékelj minket
- Profilmódosítás

A „Profilmódosítás” menüpontot választva be kell jelentkezni azzal a profillal, amit módosítani szeretnél, majd a megjelenő menüből a „becenév módosítása” vagy a „jelszómódosítás” menüpontot választva a kívánt funkció értelemszerűen teljesül.

Az „Értékelj minket” menüpontot választva a megfelelő funkcióhoz jutunk: be kell jelentkezni, be kell írni egy 1-től 5-ig terjedő egész számot, hogy hány csillagra értékeled a játékot, majd opcionálisan írhat hozzá szöveges megjegyzést is. Ezután visszatérünk a főmenübe. Fontos azonban, hogy csak akkor írhat értékelést a felhasználó, hogyha már legalább 1 játékot elkezdett.

A „Vélemények” menüpont egészen egyszerűen visszaírja a txt fájlba mentett értékelés adatokat. Az értékeléssel ellentétben itt nem kell bejelentkezni.

A „Mentett játék végeredmény kirajzolása” menüpont alatt ismét be kell jelentkezni, sikeres bejelentkezés után pedig a mentett játéktáblák között lehet lépkedni az AD gombokkal. Az adott felhasználó által elmentett játék kimenetelek mindegyike megjelenik, függetlenül a játékos aktuális játékban való szerepétől és partnerétől. Innen és az egyel korábbi funkcióból az X gomb megnyomásával lehet visszalépni a főmenübe.

„Mentett játék beolvasása” esetén mind a két játékosnak be kell jelentkeznie szerepek szerint rendben, majd a mentett játékok közül szintén az AD gombokkal lehet váltogatni, betöltés pedig az Enter megnyomására történik. Innen tovább folytatódik a játék annak rendje módja szerint.

„Új játék” esetén is két játékosnak kell bejelentkeznie szerephelyesen (először macska utána gazda), azonban itt már regisztrálhat is új felhasználó.

Regisztráció esetén be kell írni a születési dátumod yyyy:mm:dd formátumban és a nemed kiválasztani (úr, hölgy, egyéb, nem kívánom megadni; rendre 1, 2, 3 vagy 4 billentyű nyomására) (nem tárolja később), egy érvényes e-mail címet, egy egyedi becenevet és egy egyedi jelszót (utóbbit duplán), a jelszóhoz mentőkérdést kell választani melyre választ is kell adni.

Ha mindkét játékos bejelentkezett a játék sztoriját tovább tekerve létre kell hozni a pálya alakját. Ehhez meg kell adni a pálya kívánt maximális szélességét és magasságát, majd soronként, hogy mennyi mező legyen az adott sorba (ezek mindegyike páratlan egész szám kell legyen).

A játék további menete az *Általános Információk* bekezdés *Játék menete* szakasza alatt olvasható részletesen. Mindaddig, amíg az X gombbal ki nem lépünk (ilyenkor megkérdezi, akarjuk-e elmenteni a játékot) ezen az alapelven folyik tovább. Játék vége esetén egy Enter nyomása után visszatérünk a főmenübe. Ilyenkor is megkérdezik, hogy akarjuk-e menteni a végeredményt.

Bejelentkezésnél a becenév és jelszó párost kell beírni, azonban elfelejtett jelszó funkció is elérhető szükség esetén. Itt egyik lehetőségként a regisztrációkor beállított mentőkérdés helyes megválaszolásával, vagy másik lehetőségként egy, a beállított e-mail címre küldött kód helyes megadásával tudja magát azonosítani a játékos, mely után új jelszót lehet beállítani.

Hibakezelés

Felhasználó általi bemeneti hibából a felhasználó gyakorlatilag semmit nem érzékel: a helytelenül beírt adatot a program kitörli, és ugyanoda kell beírni újból egészen addig, amíg helyes adatot ír be a felhasználó

Adattárolás

Az adatok tárolását 5 db, egymással összefüggő osztály és dinamikus tömbök segítségével fogom tárolni.

Felhasználó osztály

Minden felhasználóról tároljuk a *becenevét*, *jelszavát*, mentőkérdésre adott *válaszát*, *e-mail címét*, valamint, hogy hány játékot *játszott*. Konstruktorát a globális `register()` valamint `login()` függvények hívják meg, melyek létrehozzák a játékos objektumát. Profilmódosításhoz használunk továbbá setter függvényeket is.

Önmagában egy ilyen objektum elég ahhoz, hogy azzal a játékos írni tudjon értékelést, módosítsa a profilját, vagy a saját kész játékainak végső állapotát lekérje, azonban, mivel ez egy multiplayer játék, ahol a két játékos teljesen máshogy viselkedik, így ennyi önmagában nem elég ahhoz, hogy a játékot lebonyolítsuk. Erre szolgál a **Macska** és a **Gazda** osztály, melyek miatt létre kellett hozni a jelen tárgyalt osztályba egy virtuális `lep()` tagfüggvényt, melynek jelentését az előbbi két másik osztálynál tárgyalom bővebben.

Gazda osztály

A gazda osztály megörökli a **Felhasználó** osztályt, így tehát használhatja annak minden publikus tagfüggvényét. Más egyéb privát adattagot nem kap, azonban itt definiálom a `lep()` függvényt, mely a játék során lebonyolítja a gazda lépéseit, azaz bekér egy érvényes koordinátát a játéktábláról (pl.: A3), majd ha az a mező üres, a **Mező** osztályban részletezett módon követi le a mezőre, amennyiben pedig helytelen vagy nem szabad mező koordinátája érkezik be újra kéri egészen addig, ameddig helyes koordinátát nem kap.

Macska osztály

A cica ugyanúgy megörökli a **Felhasználó** osztályt, hiszen az ő esetében is egy játékosról beszélünk, azonban mozgása és feladata jelentősen eltér a **Gazda** feladatától, ebből kifolyólag kapott új osztályt. Az öröklött adatokon kívül kap egy **Mező&** típusú privát adattagot is, mely arra a mezőre mutat a játéktáblán, amelyen a macska éppen aktuálisan áll. Ennélfogva tehát a részletezett osztályunk tartalmazza a **Mező** osztályt, hiszen ha a mező megszűnik, a macska is megszűnik.

Lépésének algoritmusai pedig a következők: először ellenőrzi, hogy éppen a szélén áll-e a táblának, illetve ezután, hogy körülötte létezik-e még olyan mező, amely szabad, ezáltal vége

van-e a játéknak. Amennyiben nem, úgy a QWEASDYX gombok valamelyikét várja a felhasználótól az *Általános Információk* bekezdés *Játék menete* szakasza alatti ábrán jelzett jelentéstartalommal. Az így meghatározni kívánt mezőt ellenőrzi, hogy szabad mező-e, amennyiben igen, úgy az aktuális helyzetét megváltoztatja a kívánt mezőre. Amennyiben nem, vagy más hibába ütközünk, újra kéri a felhasználótól egészen addig, amíg helyesen tud mozogni. A helyzetváltásban nagy segítséget jelentenek a **Mező** osztály további adattagjai.

Utolsó tagfüggvénye a kirajzol1() és kirajzol2(). Ezek rajzoló függvények, abban segítenek, hogy az *Általános Információk* bekezdésben látható karaktergrafikát ki tudja nekünk írni.

Mező osztály

Minden bizonnyal a projekt egyik legkomplexebb osztálya. Egyrészt van egy koordinátáit tároló, **Pont** típusú adattagja, tartalmazza tehát a **Pont** osztályt (ha a Pont megszűnik, a mező is). Tárol továbbá egy egyedi beszédes nevet a mezőnek a további könnyebb azonosítás érdekében. Tartalmaz ezen felül 8 db **Mező&** típusú adattagot, mely a szomszédaira mutat, amennyiben azok, illetve az aktuális mező léteznek és a játéktábla részei, vagy NULL-pointerre, amennyiben nem. Mindez a macska kényelmes mozgatásához szükséges, hiszen így a megtudott iránynak megfelelő szomszédja lesz a macska új helyzete. Minezeken kívül logikai típusú adattagként tárolja, hogy van-e rajta kő, szélén van-e, létezik-e a mező (pálya része-e), szabad-e rálépni. Ez utolsót azért kell külön emelni a kőtől, mert bár soha nem szabad, amikor kő van rajta, de nem mindig van rajta kő amikor szabad. Tagfüggvényei közé tartozik a kő lerakása (megfelelő adattag beállítása) és a macskához hasonló kirajzol1() kirajzol2() függvényei, valamint a szabad adattaghoz létezik egy setter függvény.

Az UML diagram és egy ábra a mezők láncolásáról az alábbi képeken található: (Az ábráról utólag vettem észre hogy a destruktorok lemaradtak)

