GWENT SPECIFIKÁCIÓ

Tartalomjegyzék

Bevezetés

A Gwent objektumokkal dolgozik, és a játék során ezeket az objektumokat különböző módszerekkel módosítjuk, „pakolgatjuk”. Ezek az objektumok, az ezeket kezelő szabályok, a játék levezényléséért felelős modul, és a grafikai modul is külön-külön egységekre lettek bontva a könnyebb átláthatóság és rugalmasabb fejlesztés érdekében.

Objektumok: kártyák és játékosok

kartyak.py

Ebben a modulban található a játék legkisebb építőeleme: a kártya objektumok. Itt építjük össze a paklikat, amiket átadunk a játékosnak. Továbbá itt kezeljük a kártyaállományok módosítását is: amennyiben a felhasználó törölni szeretne egy káryát, vagy újat szeretne beilleszteni, azért ez a modul felelős.

class Frakcio(Enum)

Ebben a felsorolt típusban definiáljuk a különböző frakciókat, és az egész project ez alapján hivatkozik rájuk, ezzel elkerülve, hogy a forráskód tele legyen ömlesztve sztringekkel.

class Kartya

Itt jönnek létre a kártyaobjektumok. Egy kártyának van neve, típusa és ereje, valamint egy hős-e változó (bool), mely alapján eldöntjük, érvényesek lesznek-e rá a különböző módosítások. Eltároljuk még a kártya eredeti erősségét is, mely a Clear Weather kártya letételekor lehet hasznos.

A különböző gyengítések módosításának függvényei is ebben az osztályban foglalnak helyet: időjárás effektek, és Dandelion erősítése.

def rendez

Az előadáson bemutatott kiválasztó rendezés algoritmusa: az összeállított paklit erőssége alapján rendezzük, hogy így adjuk tovább a játékos kezébe.

def osztas

Gyakorlatilag ez a kis függvény végzi el a kártyajáték lényegét: a kiválasztott frakcióból, a semleges és a speciális kártyákból összeállít véletlenszerűen – egy leosztást szimulálva – egy paklit, amit rendezve továbbít.

def pakliba_beolvas

Segédfüggvény a beolvasáshoz – úgy olvassa be a kártya adatfájlokból a sorokat, hogy azokat feldolgazva, a Kartya osztályba egyből be lehessen rakni.

def keveres

A kártyaosztás egészét vezényeli a fent leírt három függvény segítségével: a megfelelő fájlokból beolvassa a teljes kártyalistát, amit kioszthatunk a játékosnak, azokat megfelelően megkeverve továbbítja majd a játékos kezébe.

def nr_passziv

Minden frakció egyedi passzív képességgel rendelkezik, amik a játék különböző pontjain léphetnek életbe. A Northern Realms passzív képessége extra lapok kiosztását jelenti játék *közben*, ezért ezt a passzív képességet itt hajtjuk végre: kiosztunk 1 random kártyát a northern_realms adatfájlból.

def uj_adatok

Ha úgy kártyát szeretnénk hozzáadni valamely frakcióhoz vagy a semleges kártyákhoz, ez a függvény végzi annak a kártyának az adatainak a beolvasását: a neve, a típusa, valamint az ereje, mely legalább 1, legfeljebb 15 lehet.

def beolvas
Segédfüggvény új kártya hozzáadásához. Annyiban tér el a pakliba_beolvas függvénytől, hogy ez sztringként hagyja a beolvasott kártyákat a könnyebb feoldolgozás érdekében.

def hozzaad
Új kártya hozzáadását vezényeli le a fenti két függvény segítségével: amennyiben sikerül beolvasni az új kártya adatait, a meglévő kártya adatfájl felülírja oly módon, hogy az új kártyát a végére csatolja.

def torol
Ha úgy szeretnénk módosítani a kártyák adatbázisát, hogy egy meglévő elemet törlünk, azt a műveletet ez a függvény végzi el. Listázza nekünk az összes lehetséges, törölhető kártyát – tehát ami nem speciális kártya, nem speciális képességgel rendelkező kártya és nem hős kártya.

def mod
Főmenüből lehet meghívni ezt az eljárást. A kártyalista módosítását végzi el attól függően, hogy a játékos törölni vagy új kártyát szeretne beszúrni.

jatekos.py

class Jatekos
Egy játékos objektumot valósít meg: a játékosnak van neve, számon van tartva a nyert körök száma (amit ő nyert), van frakciója, és van egy paklija, melyet a kartyak.py modulból kap, egész pontosan a keveres függvény segítségével.

Nyert kör esetén az ebben az osztályban létrehozott függvény módosítja a nyert körök számát.

Grafika
A grafikai modul a Teknőcgrafikát jelenti, ez jeleníti meg a kijelzőn az „asztalt”, amin a játszma folyik. Megjeleníti a kezünkben lévő lapokat, a letett lapokat típusuk szerint kategorizálva, az ellenfél letett lapjait, szintén típusok szerint. Bal oldalt található az információs sáv, mely a meccs állását körönként frissíti. Itt található még a játékosok frakcióját szimbolizáló „ikon”.

Egy főmenüből származó, small_screen változó figyeli, ha a betűk/számok méretét kisebbre kell venni amennyiben egy kisebb felbontású kijelzőn játszunk.

grafika.py
A grafikai modul hosszúnak tűnik, azonban csupán a teknőcgrafika utasításkészlete miatt nyújtózkodott kicsit a kódsor: számos pozicionálás, 1-2 geometriai számítás és a sok frakció különböző dekorálása mind-mind apró utasítások légióját hozta létre.

def uj_ablak
A teknőc ablak a képernyőhöz igazodik, és csak a megfelelő pontokon frissítjük a képet.

def hatter, def kard, def ij, def katapult
Az alapasztal háttere, amin játszunk. A típusok beazonosítását segítik a kis ikonok, melyeket a sorok elejére rajzolunk.

def kartyat_rajzol
Ez a modul rajzolja ki a kijátszott és a még kezünkben lévő kártyákat. Kiírja a nevüket (ha túl hosszú, rövidítve teszi azt), az erejét és a típusát.

def napot/szornyet/nyilat/hajot/cimert_rajzol
Az egyes frakciók címereit lerajzoló függvények.

def mit_rajzol
Eldönti, hogy melyik frakció ikonját kell az adott helyre rajzolni.

def hatlap_rajzol
Az ellenfél kezében lévő kártyákat rajzolja csak ki: így láthatjuk, hány lapja van még ellenfelünknek.

def jatekallas_update
A bal oldalon található információs sávot rajzolja (írja) ki: jelenlegi pontszám és a nyert körök száma.

def update
Minden körben ez a függvény végzi a teljes asztal állapotának frissítését: kiszámolja hova kell rajzolni a mi lapjainkat, az ellenfél lapjait, kiíratja a játék állását. A sorban lerakott kártyákat „összeszámolva" tudja, melyiket hova kell rakni az adott sorban, illetve a kártya típusából tudja, melyik sorba kerül egyáltalán.

def bevitel
Minden körben bekéri a felhasználótó a következő lépést: egy kártya számát, vagy a passzolást. Lehetőség van még az információkat lekérdezni a 0-s gombbal, mely a konzolba kiírja a speciális kártyát képességeit.

def nyertes
Kiírja a nyertes nevét a képernyőre, vagy ha nincs ilyen, kiírja a Döntetlen feliratot.

def ki_kezd
Alapvetően nem a grafikai modul feladata eldönteni, hogy melyik játékosé a nyitó kör, azonban a Scoia'tael passzív képessége, hogy a játékos dönt erről, amit viszont a játékostól kell megkérdezni; ezért került ebbe a modulba ez a függvény.

def jatekos_nev
Beolvassuk a játékos nevét.

def frakcio_valasztas
Felsoroljuk a felhasználónak a lehetséges frakciókat, melyekből választhat saját, és ellenfél frakciót is. Lehetőség van véletlenszerű frakciók választására is.

Ellenfél taktikája, ellenfél lépései

ai.py
Ebben a modulban hoztuk létre az ellenfél (vagy adott esetben a 2. játékos) lépéseit kiszámoló függvényeket.

def lep
Ez a függvény két részre oszlik: könnyű és nehéz taktikákra. Az első négy sora összefoglalja a „könnyű" játékmód taktikáit: a gép addig pakolja le a lapokat, amíg nem ő áll jobban.

Amennyiben a nehezebb játékmód aktív, fontossági sorrendben végignéz néhány jobb, bevált, „nyerő" taktikai lehetőséget, és ezek közül a legkedvezőbbet fogja választani. Ha az ellenfélnek sok közelharci egysége van, lerak egy Biting Frostot. Ha neki van egy Dandelion lapja, úgy alakítja a játszmát, hogy biztos legyen egy kör, ahol az ő közelharci egységei könnyen megnyerhetik azt a kört a duplázott erősségek miatt. Ha van tehén, vagy O'Dimm a lapjai közt, akkor azokat azonnal kijátssza, hisz azonnali előny a következő körben. És így tovább…

A játék és a szabályai

Eddig megvannak az apró modulok, melyek felépítik a játék szerkezetét, azonban még hiányoznak a szálak, melyek mozgatják ezeket. A jatek.py és a szabalyok.py modul végzi el mindezt. A jatek.py levezényli a teljes játékmenetet elejétől a végéig, míg a különböző események forgatókönyvei a szabalyok.py modulban vannak feljegyezve.

szabalyok.py

Itt található az összes speciális kártya effektjeit elvégző függvény, valamint a játék általános szabályait definiáló függvények, röviden a játék „szabálykönyve”. Egy globális változó, a BF_aktiv bool figyeli, hogy a közelharci egységek gyengítve vannak-e, ez fontos lesz ugyanis a Dandelion kártya letétele után.

def pontszam

Ez a függvény számolja össze minden letett kártya után a játékosok aktuális pontszámát.

def asztal_torles

Ha mind a két játékos passzolt, az asztalt elő kell készíteni a következő körre. Amellett, hogy a régi lapokat eltávolítjuk, a speciális, maguk után valamit hagyó lapokat is itt vizsgáljuk, és itt aktiváljuk a képességeiket, tehát itt idézzük le a következő kör elejére a speciális lapokat.

def gyengit és def gyengitest_keres

Amennyiben egy gyengítő időjárás effekt lép érvénybe, ezek a függvények végzik el a megfelelő kártyák gyengítését. A gyengit fv maga a gyengítést végzi el, a gyengitest_keres pedig vizsgálja, mely kártyákon melyik gyengítést kell végrehajtani.

def clear_weather

A Clear Weather kártya eltávolítja az időjárást torzító kártyákat, és a kártyák erejét visszaállítja: ez a függvény ezt a műveletet végzi el.

def scorch, def eleget, def legerosebbet_keres

A scorch kártya három függvénye. A scrorch kártya elégeti az asztalon található legerősebb, nem-hős kártyákat. A legerosebbet_keres megkeresi azokat a kártyákat, melyekre érvényesül ez a kritérium. Ezt a függvényt a scorch függvény hívja meg, amely, miután megkapta a maximum értéket, elégeti a megfelelő kártyákat az eleget függvénnyel.

def specialis_effektek

Minden körben végignézi a letett kártyákat, amennyiben van új speciális effekt, annak megfelelően hívja meg a fent definiált, speciális effekteket elvégző függvényeket.

def dandelion, def villmerth

Dandelion, és Villmerth kártyák effektjeit hajtják végre. Dandelion megerősíti az összes nem-hős közelharci egységet, amennyiben nincs aktív gyengítés rajtuk – ebben segít a BF_aktiv változó.

Villmerth pedig elégeti az ellenfél legerősebb, nem-hős közelharci egységét (vagy egységeit, amennyiben többen rendelkeznek ugyan azzal az erőponttal).

def buff_kartyak

Amennyiben talál Dandelion vagy Villmerth kártyákat, a fenti függvényeket hívja meg. Amennyiben a monster deck egyik Crone-ját raktuk le, lerakunk mellé automatikusan egy 6-os erősségű Fiend-et.

def info

Az infó adatfájl szövegét vetíti ki a konzolra: felsorolja a játékosnak, hogy melyik speciális kártya milyen képességekkel rendelkezik. Ezt bármely körben megtehetjük.

jatek.py

A játék legelejétől legvégéig ez a modul vezényeli az eseményeket.

def jatek

Ez a függvény végzi a játék inicializálását (nevek, frakciók), majd meghívja az eredményhirdető eljárást, a játszma függvény eredménye alapján.

def jatszma

Az inicializálás során létrehozott adatok alapján készít két játékos objektumot. Majd ezzel a két játékossal lefut egy while ciklusban az egész játszma. Minden körben egy interakció megengedett, ezen interakció alapján változik a játék tábla (új kártya, új kör, új effektek, stb). A körök közt 1-1 másodperc szünet jön, hogy legyen időnk megszemlélni, átgondolni a lépéseket.

A grafikai egység frissül a kör elején, majd bekérjük a játékostól (vagy géptől), hogy mely kártyát szeretné letenni. A kártya letétele után a szabalyok modul megvizsgálja, milyen módosításokat kell elvégezni a táblán.

Amikor valamelyik játékos megnyeri a második körét, kiugrunk a ciklusból – játék vége – és jön az eredményhirdetés. Ennek az eljárásnak a visszatérési értéke lesz a győztes neve, vagy None, ha döntetlen.

def eredmeny

Az előbb kiértékelt nyertes neve alapján hívja meg a grafikai modul eredményhirdető függvényét. Eredményhirdetés után három másodperccel bezáródik az ablak, visszaugrunk a főmenübe, konzolra.

Főmenü, a menüvezérelt program

main.py

A main modulban definiálunk néhány globális változót (kis kijelző, két gép, nehézség), majd a főmenü egy végtelen ciklusban vezényeli a játék különböző pontjait. Új játék, nehézség állítás, gép a gép ellen, kijelző méretének beállítása, amire lehetőségünk van. Kilépésre nyomás után kilépünk a ciklusból, a program bezárul.