VERES BENEDEK ZOLTÁN – SJG, KOMÁROM
INFOPROG 2024

KOCKAPÓKER

Tartalom:

I. Instaláció

Első lépésben nyomjunk rá a setup.bat nevezetű fájlra, ez létrehoz a virtuális környezet és felinstalálja a szükséges könyvtárakat (ehhez internet szükséges).

Második lépésben indítsuk el a játékot, a továbbiakban is, a start.bat fájllal

II. Használati útmutató

- a játékban alapvetően 7 billentyűt használunk:

- **Fel** - fel nyilacska
- **Le** - le nyilacska
- **Balra** - balra nyilacska
- **Jobbra** - jobbra nyilacska
- **Belép/elfogad/érték választása** - enter
- **visszalépés** – backspace

III. A játékmenete

a) Játék bekapcsolásakor az üdvözlőlap fogad, ahol meg kell adnunk a nevünket (üres és "CPU" nem lehet), majd kiválasztani a nehézségi fokozatot(itt a balra-jobbra gombokkal mehetünk). Mindkét esetben az enter gombbal fogadjuk el a választásunkat

b) 2. **Főmenü** : Itt több lehetőségünk van

   a. Folytatatás - ha van egy játék mentésünk, akkor azt betölti, !figyelem betöltés után a fájlt kitörli, így legközelebb újra el kell menteni

   b. Új játék - új játék kezdése

   c. Rangsor - kiírja a highscores.txt fájlból a legmagass pontszámokat és, hogy ki nyert(CPU a gép), a backspace billentyűvel léphetünk vissza a főmenübe

   d. Beállítások - a név és a nehézség állítható be itt

c) 3. **Játék menete**

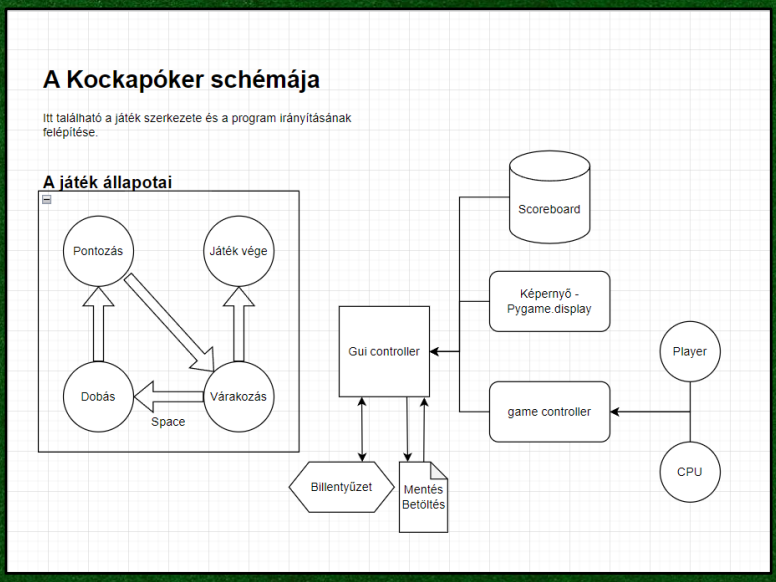
   1) A játékos kezd
   2) DOBÁS - A space megnyomása után szimuláljuk a dobást, kiértékeli a dobást és kiírja a lehetséges pontszámokat a táblázatba
   3) PONTVÁLASZTÁS - A fel-le billentyűkkel mehetünk, majd az enter billentyűvel kiválasztjuk a még nem kiválasztott pontszámok közül, amelyiket szeretnénk
   4) GÉP - A space megnyomása után dob a gép, kiértékeli

5) GÉP PONTOZÁS - A nehézségtől függően a gép kiválasztja a megfelelő értéket: - könnyű - kiválasztja az első értéket, amely nem 0 (ha lehetséges - nehéz - kiválasztja a lokális maximumot
6) (Ha mind a 18 dobás lement) - JÁTÉK VÉGE, (másképp) - vissza a második lépéshez

IV. A program felépítése

A program megírás közben törekedtem, hogy minnél modulárisabb legyen. Ez azt okozta, hogy elég sok alprogram épül fel. Ez tette lehetővé, hogy a program legtöbb részéhez izoláltan hozzá lehessen nyúlni, ezzel hozzá segítve a hatékonyabb fejlesztéshez. A program infrastruktúrája 5+1 fő komponensből áll:

1) Game controller – játék vezérlő - a játék logikáját foglalja magába, és annak irányítása
2) Képernyő – ezen keresztül rajzoluk ki az alakzatokat a képernyőre
3) Scoreboard – pontozó tábla – a meccs közbeni pontok vannak itt feltüntetve, tárolás és render egyben
4) PlayerEntity – alap osztály, amelyből létrejön a Player és a CPU osztályok
5) GUI controller – összes többi modult fogja össze és egy program vezérlőként funkcionál
6) Fő program



1. Game controller - Játékvezérlő

Lehetővé teszi, hogy váltogassuk az aktív játékosokat, előkészíti az adatokat a mentéshet, és a beolvasott JSON-t átalakítja

2. Képernyő

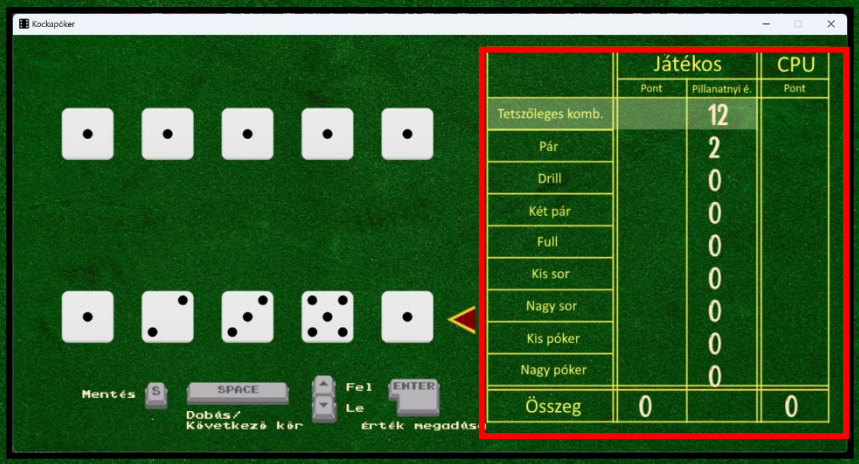
A pygame display objuktumát használom erre, az végzi el a piszkos munkát ezzel kapcsolatban

3. Scoreboard – pontozó tábla

A pontozó táblán (alábbi képen piros kerettel kijelöl rész) 3 oszlop található: Játékosnak 2 CPU-nak 1.

A játékos Pillantnyi érték nevezetű oszlopában, azokat az értékeket számolja ki a program, amelyeket dobás után az adott kombinációk érnek. Miután a játékos kiválasztotta, melyik kombinációt akarja, ezután a pont átíródik a *Pont* oszlopba.

A CPU oszlopban a gép választott pontszáma található, mivel a gép maga számolja ki és választja ki a számára megfelelő pontszámot, ezért itt nincs szükség még egy oszlopra



4. PlayerEntity – Player, CPU

Ebben a részben található a PlayerEntity törzsosztály, melyben az alapvető változók, maga a dobás tárolása, a lehetséges kombinációk itt találhatóak (pár, drill, kis póker, nagy póker,...), valamint itt találhatóak a mindkét leányosztáy számára alapvető metódusok mint az új dobás szimulálása, a dobás értékelése.

A Player osztály a PlayerEntity leányosztálya személyre szabja a játkosra, itt taláható a mentés és beolvasás részek a játékos számára

A CPU osztály szintúgy a PlayerEntity leányosztáya, a előbbieken kívül itt található a play_hand metóduso, melyben a két nehézségi fokozat szerint lép a gép.

A pontozó rendszert Test Driven Development(TDD) elvei alapján fejlesztettem le a legbiztosabb működésért, ezért található a modulok almappában egy tests mappa, amelyben fellelhetőek a tesztek a különböző dobás kombinációkra figyelve az éllesetekre körülbelül 59 teszt található itt. Igyekeztem az értékeléset a legnagyobb átláthatóság kedvéért a legegyszerűbben, legolvashatóbban megírni, betartva python konvenciáit.

```python
@staticmethod
def full(dobas_data: DobasData) -> int:
    return sum(dobas_data.dobas_sor) if dobas_data.counter_2 == {2: 1, 3: 1} else 0

9 usages  Dewliak
@staticmethod
def kis_poker(dobas_data: DobasData) -> int:
    leggyakoribb_szam = max(dobas_data.counter_1, key=dobas_data.counter_1.get)
    return 4 * leggyakoribb_szam if dobas_data.counter_1[leggyakoribb_szam] >= 4 else 0

9 usages  Dewliak
@staticmethod
def tetszoleges_kombinacio(dobas_data: DobasData) -> int:
    return sum(dobas_data.dobas_sor)

9 usages  Dewliak
@staticmethod
def par(dobas_data: DobasData) -> int:
    parok = list(filter(lambda elem: elem[1] >= 2, dobas_data.counter_1.items()))

    if len(parok) == 0:
        return 0

    return 2 * max(parok)[0]

9 usages  Dewliak
@staticmethod
def drill(dobas_data: DobasData) -> int:
    harmasok = list(filter(lambda elem: elem[1] >= 3, dobas_data.counter_1.items()))

    if len(harmasok) == 0:
        return 0

    return 3 * max(harmasok)[0]
```

A kocka dobások szimulálásához létrehoz egy saját típust dataclass segítségével (a programban DobasData névvel), ebben összpontosítottam mindent, ami a kockák szimulálásához szükséges (logikai szempontból, a grafikai rész nem itt található). A DobasData-ban megtalálható maga a dobás egy tömbként, amelyben 5 szám található a dobások száma 1 – 6 -ig, dobas_sor néven található. Található még két darab counter típus, amit a python beépített collection könyvtára, az első counter össze számolja, miből mennyi van. Ennek segítségével találom meg például a párokat vagy a drillt. Egy példában a dobás = [2, 2, 3, 5, 3], az első counterben ezek az értékek találhatóak {2: 2, 3: 2, 5:1}, ha végig megyünk rajta láthatjuk a 2-esből és a 3-asból 2 darab van, így a párokat kitudjuk számolni. A második counter az első counter-e ,ez arra jó, hogy megtaláljuk hány darab pár van, a dupla párokhoz például. Folytatva az előző példát itt ezeket az értékeket kapjuk {1: 1, 2: 2} tehát egy darab 1-es van és két darab pár, így ez egy dupla pár is egyben.

5. GUI controller

Ez a modul fogja össze az összes többi modult irányít és komunikál a többi modul között. A legfőbb oka ennek a modulnak, hogy elég nehéz megoldani másképpen a különböző python fájlok, a modulok, közötti kommunikációt, megosztani a megf elelő változókat. Ez az osztály ezt mind egyszemélyben és központilag megoldja. Ez a rész még egy fontos dolgot magába foglal, amiről eddigiekben még nem volt szó az pedig a billentyűzet gombjainak lenyomásának érzékelése.

6. Fő program

Ebben a részben igazából betöltöm az összes fontos modult és könyvtárat és váltogat a különböztő állapotok között, például menü állapot, játék állapot,...