Programozói dokumentáció

1. A projekt felépítése

Ez a projekt egy egyszerű dobógép szimulációt valósít meg Python nyelven. Az alábbi fájlokat tartalmazza:

- UI.py
- sampler_functions.py
- rythm_nl_functions.py
- save_load.py
- main.py

A futtatáshoz szükséges környezet:

- Python 3.9 vagy újabb verzió
- Nincs szükség külső könyvtárak telepítésére, mivel a kód kizárólag a Python szabványos könyvtárait használja.

2. Adatszerkezetek és tervezési megfontolások

A programban használt adatszerkezetek, mint például szótárak, listák és osztályok, kulcsszerepet játszanak az alkalmazás működésében. Ezek az adatszerkezetek lehetővé teszik a dinamikus adatkezelést, a program állapotainak nyomon követését, valamint a felhasználói interakciók kezelését. Az alábbiakban bemutatom az egyes adatszerkezetek szerepét és azok tervezési megfontolásait.

1. Listák

A listák olyan adatszerkezetek, amelyek egyszerűen tárolják az elemeket, és lehetővé teszik azok gyors hozzáférését és módosítását. A programban a listák különböző elemeket tartalmaznak, például gombokat, mintákat és egyéb grafikai komponenseket.

Példa:

- **rythm_nl_menu_buttons** és **small_circle_buttons**: Ezek listák, amelyek gombokat és vizuális elemeket tartalmaznak, amelyeket a felhasználó kattintásai alapján módosítanak. A gombok változása és a felhasználói interakciók figyelése érdekében szükségesek ezek a listák, hogy a program dinamikusan tudjon reagálni.
- **relative_primes**: Ez a lista tartalmazza a ritmusnyaklánc relatív prímszámait, amelyeket a program a minták lejátszása során használ.

Tervezési megfontolás:

- A listák használata lehetővé teszi az elemek könnyű hozzáadását és eltávolítását a futásidő alatt. Az ilyen dinamikusan változó elemek esetében a listák a legmegfelelőbbek.
- A listák segítenek abban, hogy a különböző komponensek (például gombok vagy ritmusnyaklánc elemek) könnyen manipulálhatók és átrendezhetők legyenek.

2. Szótárak

A szótárak olyan adatszerkezetek, amelyek kulcs-érték párokat tárolnak. A programban szótárakat használunk, hogy a komponensek közötti kapcsolatokat és állapotokat tároljuk.

Példa:

- A **sampler_buttons** és **sampler_menu_buttons** változók szótárak lehetnek, amelyek kulcs-érték párokban tárolják a gombokhoz tartozó adatokat (például állapotok, viselkedések). A gombok állapota alapján az alkalmazás dinamikusan változtatja a vizuális megjelenést és a műveletek működését.

Tervezési megfontolás:

- A szótárak használata gyors hozzáférést biztosít a különböző komponensekhez a kulcsok alapján, így az alkalmazás állapotának kezeléséhez és a felhasználói interakciókhoz is hatékonyan alkalmazható.
- A szótárak segítenek abban, hogy a gombok vagy egyéb elemek hozzáférhetők és módosíthatók legyenek anélkül, hogy az adatokat lineáris módon kellene keresni.

3. Osztályok

Az osztályok lehetővé teszik az objektumorientált megközelítés alkalmazását, ahol az egyes komponensek és azok viselkedése összekapcsolódik. A programban használt osztályok a különböző vizuális és logikai komponensek modellezésére szolgálnak.

Példa:

- Az **input_box_button** osztály az egyik példája, amely egy gombot reprezentál, amely lehetővé teszi a felhasználó számára, hogy szöveget adjon meg. Az osztály tárolja a gomb pozícióját, állapotát és egyéb fontos adatokat.
- Az osztályok a gombok dinamikus kezelését és a felhasználói interakciók kezelését is megkönnyítik.

Tervezési megfontolás:

- Az osztályok segítenek a kód moduláris és karbantarthatóbbá tételében. A különböző vizuális elemeket (pl. gombok) és azok viselkedését osztályokba szervezve könnyebben karbantarthatók és módosíthatók a program különböző részei.
- Az osztályok lehetővé teszik a kód újrahasználhatóságát is, mivel a komponensek egyes osztályokba szervezve könnyen más projektekben is felhasználhatók.

Tervezési Megfontolások

A fentiek alapján az alkalmazás tervezésénél figyelembe vettük az alábbi szempontokat:

1. Modularitás

Az adatszerkezetek (listák, szótárak, osztályok) segítenek abban, hogy az alkalmazás különböző részei elkülönüljenek egymástól. A kód átláthatóbbá válik, és a különböző funkciók könnyebben módosíthatók anélkül, hogy a többi részt is érintené.

2. Dinamikus Adatkezelés

A listák és szótárak lehetővé teszik a futásidő alatt történő dinamikus adatkezelést. A felhasználói interakciók, mint például kattintások vagy billentyűleütések, hatással vannak az adatok állapotára, és a programnak képesnek kell lennie arra, hogy ezeket az adatokat gyorsan frissítse és újra felhasználja.

3. Karbantarthatóság

A különböző adatszerkezetek használata lehetővé teszi, hogy a program könnyen bővíthető legyen. Például ha új gombokat vagy mintákat szeretnénk hozzáadni, elegendő új elemeket hozzáadni a megfelelő listákhoz vagy szótárakhoz.

4. Skálázhatóság

Az osztályok és az adatszerkezetek használata biztosítja, hogy az alkalmazás skálázható legyen. Ha további funkciókat vagy komponenseket kell hozzáadnunk, akkor a meglévő struktúrák könnyen kiegészíthetők, anélkül, hogy az alapfunkciókat megzavarnánk.

5. Felhasználói élmény

Az alkalmazásban alkalmazott adatszerkezetek (pl. gombok kezelése, minták lejátszása) lehetővé teszik a felhasználó számára, hogy intuitív módon interakcióba lépjen a programmal. Az adatok gyors kezelése biztosítja a sima felhasználói élményt, minimalizálva a késéseket vagy a hibás működést.

Ezek a tervezési megfontolások segítik az alkalmazás hatékony működését és a könnyű karbantartást, miközben biztosítják a dinamikus adatkezelést és a felhasználói interakciók gyors és pontos reagálását.

3. Függvények dokumentációja

Fájl: UI.py

1. Button osztály

- **Feladat:**
  Gomb objektum definiálása, amely interakciók kezelésére szolgál.
- **Attribútumok:**
  - x (int): A gomb X koordinátája.
  - y (int): A gomb Y koordinátája.
  - active (bool): Jelzi, hogy a gomb aktív-e.
  - width (int): A gomb szélessége.
  - height (int): A gomb magassága.
  - circle_indices (list, optional): A gombhoz rendelt körök indexei.
  - color (tuple, optional): A gomb színe (alapértelmezett: (94, 80, 63)).
  - file_name_text (str, optional): A gombhoz kapcsolt fájl neve (alapértelmezett: üres).

---

2. Circle osztály

- **Feladat:**
  Kör objektum definiálása, amely a ritmus nyaklánc elemeit reprezentálja.
- **Attribútumok:**
  - x (int): A kör X koordinátája.
  - y (int): A kör Y koordinátája.
  - active (bool): Jelzi, hogy a kör aktív-e.
  - radius (int): A kör sugara.

---

3. create_screen()

- **Feladat:**
  Létrehozza az alkalmazás képernyőjét.
- **Visszatér:**
  - screen (objektum): A Pygame képernyő objektuma.

---

4. sampler_position(screen, width, height, buttons)

- **Feladat:**
  Megjeleníti a sampler gombokat egy meghatározott területen.
- **Paraméterek:**
  - screen (objektum): A megjelenítő képernyő.
  - width (int): A sampler X pozíciója.
  - height (int): A sampler Y pozíciója.
  - buttons (list): A samplerhez tartozó gombok.
- **Visszatér:**
  - buttons (list): A sampler gombjainak frissített listája.

---

5. sampler_menu(screen, width, height, menu_buttons, sampler_buttons,

input_box_button, sampler_button_index, mouse_pos=None, click=False)

- **Feladat:**
  Kezeli a sampler menü gombjait és azok megjelenítését.
- **Paraméterek:**
  - screen (objektum): A megjelenítő képernyő.
  - width (int): A menü X pozíciója.
  - height (int): A menü Y pozíciója.
  - menu_buttons (list): A menü gombjai.

- sampler_buttons (list): A samplerhez tartozó gombok.
- input_box_button (objektum): Az input mező gombja.
- sampler_button_index (int): Az aktuálisan aktív sampler gomb indexe.
- mouse_pos (tuple, optional): Az egér aktuális pozíciója.
- click (bool, optional): Jelzi, hogy történt-e kattintás.

- **Visszatér:**
  - menu_buttons (list): A frissített menü gombjai.

---

6. rythm_circle(screen, width, height)

- **Feladat:**
  Megrajzolja a ritmus nyakláncot reprezentáló fő kört.
- **Paraméterek:**
  - screen (objektum): A megjelenítő képernyő.
  - width (int): A kör középpontjának X koordinátája.
  - height (int): A kör középpontjának Y koordinátája.
- **Visszatér:**
  - Nincs.

---

7. rythm_necklace_menu(screen, width, height, buttons, step_input,

chosen_relative_prime, mouse_pos=None, click=False)

- **Feladat:**
  Létrehozza és kezeli a ritmus nyaklánc menüjét, amely a lépésszámot, eseményeket, valamint egyéb funkciókat jelenít meg.
- **Paraméterek:**
  - screen (objektum): A megjelenítő képernyő.
  - width (int): A menü X pozíciója.

    - height (int): A menü Y pozíciója.
    - buttons (list): A menü gombjainak listája.
    - step_input (int): A lépések száma.
    - chosen_relative_prime (int): A kiválasztott relatív prímek száma.
    - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
    - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
    - Nincs.

Fájl: sampler_functions.py

1. empty_sampler_button_gen(width, height)

- **Feladat:**
  Létrehoz egy 2x3-as gombmátrixot a sampler számára.
- **Paraméterek:**
    - width (int): A gomb szélessége.
    - height (int): A gomb magassága.
- **Visszatér:**
    - buttons (list): Egy listát ad vissza, amely a gombobjektumokat (UI.Button) tartalmazza.

---

2. existing_sampler_buttons(screen, sampler_buttons, small_circle_buttons, mouse_pos=None, click=False)

- **Feladat:**
  Frissíti a sampler gombjainak állapotát és megjelenését a felhasználói interakciók alapján.
- **Paraméterek:**
    - screen (objektum): A megjelenítő képernyő.
    - sampler_buttons (list): A samplerhez tartozó gombok listája.
    - small_circle_buttons (list): A kicsi körgombok listája.
    - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
    - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
    - Nincs.

---

3. sampler_menu_button_gen(width, height)

- **Feladat:**
  Létrehoz három menügombot a sampler menü számára.
- **Paraméterek:**
  - width (int): A gomb szélessége.
  - height (int): A gomb magassága.
- **Visszatér:**
  - buttons (list): A menügombokat tartalmazó lista.

---

4. input_box_button_gen(second_sampler_menu_button)

- **Feladat:**
  Létrehoz egy gombot a bemeneti szövegdoboz számára.
- **Paraméterek:**
  - second_sampler_menu_button (objektum): A második menügomb, amelyhez igazítja a szövegdobozt.
- **Visszatér:**
  - button (objektum): A létrehozott szövegdoboz gomb.

---

5. active_sampler_button(sampler_buttons)

- **Feladat:**
  Visszaadja az aktív sampler gomb indexét.
- **Paraméterek:**
  - sampler_buttons (list): A sampler gombjainak listája.
- **Visszatér:**
  - active_button_index (int or None): Az aktív gomb indexe, vagy None, ha nincs aktív gomb.

---

6. input_box(screen, input_box_button, sampler_button_index, sampler_buttons, mouse_pos=None, click=False)

- **Feladat:**
  Kezeli a bemeneti szövegdoboz megjelenítését és állapotát.
- **Paraméterek:**
  - screen (objektum): A megjelenítő képernyő.
  - input_box_button (objektum): A szövegdoboz gomb.

    - sampler_button_index (int): Az aktív sampler gomb indexe.
    - sampler_buttons (list): A sampler gombjainak listája.
    - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
    - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
    - Nincs.

---

7. active_upload_button_message(screen, sampler_menu_button)

- **Feladat:**
  Megjeleníti az üzenetet, amely a fájl feltöltésére szólítja fel a felhasználót.
- **Paraméterek:**
    - screen (objektum): A megjelenítő képernyő.
    - sampler_menu_button (objektum): A menügomb, amelyhez az üzenet tartozik.
- **Visszatér:**
    - upload_butt_text (str): Az üzenet szövege.

---

8. play_button(sampler_buttons, sampler_menu_buttons, mouse_pos=None, click=False)

- **Feladat:**
  Lejátszik egy fájlt az aktív sampler gombhoz rendelve, ha a fájl elérhető.
- **Paraméterek:**
    - sampler_buttons (list): A sampler gombjainak listája.
    - sampler_menu_buttons (objektum): A lejátszásért felelős menügomb.
    - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
    - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
    - Nincs.

---

9. delete_button(sampler_buttons, sampler_menu_buttons, mouse_pos=None, click=False)

- **Feladat:**
  Törli az aktív sampler gombhoz rendelt fájl nevét és indexeit.
- **Paraméterek:**
    - sampler_buttons (list): A sampler gombjainak listája.
    - sampler_menu_buttons (objektum): A törlésért felelős menügomb.
    - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
    - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
    - Nincs.

Fájl: rythm_nl_functions.py

1. step_count(step_num, mouse_pos, click, button)

- **Feladat:**
  Növeli vagy csökkenti a lépések számát egy interakciós gomb segítségével.
- **Paraméterek:**
  - step_num (int): Az aktuális lépésszám.
  - mouse_pos (tuple): Az egér aktuális pozíciója.
  - click (bool): Jelzi, hogy történt-e kattintás.
  - button (list): A lépésszám változtatásához használt gomb(ok).
- **Visszatér:**
  - step_num (int): A módosított lépésszám.

---

2. is_being_divisor(step_number, divisors)

- **Feladat:**
  Ellenőrzi, hogy egy adott szám nem osztható-e megadott osztók egyikével sem.
- **Paraméterek:**
  - step_number (int): A vizsgált szám.
  - divisors (list): Az osztók listája.
- **Visszatér:**
  - bool: True, ha a szám nem osztható egyik osztóval sem, különben False.

---

3. relative_primes(step_number)

- **Feladat:**
  Megadja az adott szám relatív prímjeinek listáját.
- **Paraméterek:**

- step_number (int): A vizsgált szám.

- **Visszatér:**
  - relative_primes (list): Az adott szám relatív prímjeinek listája.

---

4. relative_prime_index(step_number, relative_prime_count_index, button, mouse_pos, click)

- **Feladat:**
  Frissíti a relatív prímek indexét a felhasználói interakció alapján.
- **Paraméterek:**
  - step_number (int): Az aktuális lépésszám.
  - relative_prime_count_index (int): A relatív prímek aktuális indexe.
  - button (objektum): A relatív prímek indexét szabályozó gomb.
  - mouse_pos (tuple): Az egér aktuális pozíciója.
  - click (bool): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
  - index (int): A módosított relatív prím index.

---

5. r_necklace_menu_button_create(width, height)

- **Feladat:**
  Létrehoz egy öt gombból álló menüt a ritmus nyaklánc funkcióhoz.
- **Paraméterek:**
  - width (int): A gombok szélessége.
  - height (int): A gombok magassága.
- **Visszatér:**
  - buttons (list): Az öt gombot tartalmazó lista.

---

6. small_rythm_circles_button_gen(step_number, width, height)

- **Feladat:**
  Generálja a kis köröket, amelyek a ritmus nyakláncot ábrázolják.
- **Paraméterek:**
  - step_number (int): A kis körök száma.
  - width (int): A nyaklánc középpontjának X koordinátája.
  - height (int): A nyaklánc középpontjának Y koordinátája.
- **Visszatér:**
  - circles (list): A kis körök objektumainak listája.

---

7. existing_small_rythm_circles(screen, circles, chosen_relative_prime, mouse_pos=None, click=False)

- **Feladat:**
  Kezeli a ritmus nyaklánc kis köreinek interakcióját és megjelenítését.
- **Paraméterek:**
  - screen (objektum): A megjelenítő képernyő.
  - circles (list): A kis köröket reprezentáló objektumok listája.
  - chosen_relative_prime (int): A kiválasztott relatív prímek száma.
  - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
  - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
  - Nincs.

---

8. active_event_circle_button(circles, sampler_buttons)

- **Feladat:**
  Rögzíti az aktív körök indexeit az aktuális sampler gombhoz.

- **Paraméterek:**
    - circles (list): A kis körök listája.
    - sampler_buttons (list): A samplerhez tartozó gombok listája.
- **Visszatér:**
    - Nincs.

---

9. start_button(width, height, is_playing, mouse_pos=None, click=False)

- **Feladat:**
  Indítja vagy megállítja a lejátszást a "Start" gomb használatával.
- **Paraméterek:**
    - width (int): A menü szélessége.
    - height (int): A menü magassága.
    - is_playing (bool): Jelzi, hogy éppen zajlik-e lejátszás.
    - mouse_pos (tuple, optional): Az egér aktuális pozíciója.
    - click (bool, optional): Jelzi, hogy történt-e kattintás.
- **Visszatér:**
    - is_playing (bool): A lejátszás új állapota.

---

10. note_playing(note_count, sampler_buttons, step_number)

- **Feladat:**
  Lejátssza a megfelelő hangjegyet a sampler gombok alapján az aktuális lépés során.
- **Paraméterek:**
    - note_count (int): Az aktuális hangjegy száma.
    - sampler_buttons (list): A samplerhez tartozó gombok listája.
    - step_number (int): A lépések száma.
- **Visszatér:**

- note_count (int): A frissített hangjegy szám.

Fájl: save_load.py

1. save_button(sampler_buttons, step_number)

- **Feladat:**
  A sampler gombok aktuális állapotát és a lépésszámot elmenti egy fájlba (save.dat).
- **Paraméterek:**
  - sampler_buttons (list): A sampler gombok listája, amelyek állapotát el kell menteni.
    - A gombok tartalmazzák a következő attribútumokat:
      - file_name_text (str): A gombhoz rendelt fájlnév.
      - circle_indices (list): A gombhoz társított aktív körök indexei.
  - step_number (int): A lépések száma, amely az aktuális konfigurációt jellemzi.
- **Fájl:**
  - A mentés a következő struktúrával történik a save.dat fájlban:

1. Az első sor tartalmazza a lépések számát.
2. Minden gombhoz két sor tartozik:
   - Az első a file_name_text.
   - A második a circle_indices listát tartalmazza.

- **Visszatér:**
  - Nincs.

---

2. load_button(sampler_buttons)

- **Feladat:**
  Betölti a mentett állapotot a save.dat fájlból, és frissíti a sampler gombokat és lépésszámot.

- **Paraméterek:**
    - sampler_buttons (list): A sampler gombok listája, amelyek állapotát frissíti a mentett adatok alapján.
- **Visszatér:**
    - step_number (int): A fájlból betöltött lépésszám.
- **Kivétel:**
    - FileNotFoundError: Ha a save.dat fájl nem található, a függvény hibát dob.
- **Működés:**

1. Ellenőrzi, hogy a save.dat fájl létezik-e.
2. Betölti a lépésszámot az első sorból.
3. A további sorokat páronként olvassa:
    - Az első a gombhoz tartozó fájlnevet (file_name_text) adja meg.
    - A második a körök indexeinek listáját (circle_indices) tartalmazza.

Fájl: main.py

Ez a függvény a program belépési pontja. A pygame könyvtár segítségével inicializálja a szükséges komponenseket és beállítja a fő ciklust.

- **Feladat:**
    - Inicializálja a pygame környezetet.
    - Kezeli az eseményeket (kattintások, billentyűleütések).
    - Frissíti és kirajzolja a képernyőt.
    - Felhasználói interakciókat figyeli és azokat megfelelően kezeli (pl. gombnyomások, minta lépések, stb.).
    - Mentést és betöltést végez el, ha a megfelelő gombok megnyomásra kerülnek.
- **Változók és Funkciók:**

- **pygame.init()** és **pygame.mixer.init()**: Inicializálja a pygame könyvtárat és a hangkeverőt.
- **PLAYBACK_EVENT**: Egyedi esemény típus a minták lejátszásához.
- **clock**: A játék ciklus sebességét szabályozza.
- **screen**: A pygame ablak (képernyő) létrehozása.
- **step_number**: A lépések száma a ritmus nyakláncban.
- **relative_prime_count_index**: Az index a relatív prímszámok listájában.
- **note_count**: Nyilvántartja, hogy hány hangot játszott le eddig.
- **rythm_nl_menu_buttons**: A ritmus nyaklánc menü gombjainak listája.
- **relative_primes**: A relatív prímszámok listája.
- **small_circle_buttons**: A kis körökhöz tartozó gombok listája.
- **sampler_buttons**: A sampler gombok listája.
- **sampler_menu_buttons**: A sampler menü gombok listája.
- **input_box_button**: A szövegdoboz gombja.
- **last_sampler_button_index**: A legutóbbi kiválasztott sampler gomb indexe.

Események kezelése

A main() függvényben különböző események történhetnek, amelyeket az alábbiakban mutatok be:

1. Quit (Kilépés)

- **Esemény:** Ha a felhasználó bezárja az ablakot (pygame.QUIT).
- **Hatás:** A run változót False-ra állítja, kilép a ciklusból.

2. Kattintás (MOUSEBUTTONDOWN)

- **Esemény:** Ha a felhasználó rákattint az egér bal gombjával.
- **Hatás:** Beállítja a click változót igazra, ami jelzi, hogy történt kattintás.

3. Lejátszás (PLAYBACK_EVENT)

- **Esemény:** Az egyéni esemény, amelyet a ritmusnyaklánc elkezd játszani.
- **Hatás:** A note_count értéke frissül, és a következő hangot játssza le a sampler gombokból.

Interakciók kezelése

A felhasználó által végzett interakciók, mint például gombnyomások, mezők frissítése, stb. a következő módokon történnek:

1. Sampler gombok

- A sampler_functions.existing_sampler_buttons és sampler_functions.active_sampler_button függvények frissítik és vizsgálják, hogy a felhasználó rákattintott-e valamelyik gombra.
- A UI.sampler_menu függvény kezeli a sampler menü kirajzolását és frissítését.

2. Szövegbevitel (Input Box)

- Ha a felhasználó gépel, a event.key értékétől függően frissíti a szöveget a sampler gombokon, majd a változást visszajelzi a képernyőn.

3. Ritmikus Nyaklánc gombok

- A rythm_nl_menu_buttons elemeken végzett kattintások a ritmus nyaklánc lépéseit szabályozzák, mint például a start_button, save_button, és load_button gombok.

Mentés és Betöltés

- **Mentés:** A save_load.save_button függvény menti a jelenlegi állapotot egy fájlba.
- **Betöltés:** A save_load.load_button függvény betölti a mentett állapotot.

Mentés és Betöltés kezelése:

- A save_button akkor aktiválódik, amikor a felhasználó rákattint a mentés gombra.
- A load_button akkor aktiválódik, amikor a felhasználó rákattint a betöltés gombra, és visszaállítja a lépésszámot és a gombok állapotát.

Fő Ciklus és Frissítés

A fő ciklus minden egyes iterációjában a következő történik:

- Események kezelése.
- A képernyő frissítése és újrarajzolása.
- A felhasználói interakciók figyelése és megfelelő függvények meghívása.

A képernyő minden 60 Hz-es frissítéssel frissül (ez a clock.tick(60) hívással történik).