

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»  
Кафедра «Комп'ютерні інформаційні технології»

**Пояснювальна записка**  
до кваліфікаційної роботи бакалавра

на тему: «Розробка застосунку «Музична школа»: тренажер музичного слуху»


за освітньою програмою: «12 Інженерія програмного забезпечення»

зі спеціальності: «121 Інженерія програмного забезпечення»

Виконав: студент групи «ПЗ1912»

Керівник:

Нормоконтролер:

  
\_\_\_\_\_  
(підпис студента)

/Максим ЛИСЕНКО/

(Ім'я ПРІЗВИЩЕ)

  
\_\_\_\_\_  
(підпис)

/ст. викл. Ірина ШАПОВАЛ/

(посада, Ім'я ПРІЗВИЩЕ)


  
\_\_\_\_\_  
(підпис)

/Світлана ВОЛКОВА/

(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з  
праць інших авторів без відповідних посилань.

Студент

  
\_\_\_\_\_  
(підпис)

Дніпро – 2023 рік

Ministry of Education and Science of Ukraine  
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»

Department «Computer information technology»

Explanatory Note  
to Bachelor's Thesis

on the topic: « Development of the «Music School» application: musical hearing  
simulators»

according to educational curriculum « Software engineering »

in the Speciality: «121 Software engineering»

Done by the student of the group PZ:

//

Scientific Supervisor:

/Iryna SHAPOVAL/

Normative controller:

//

Dnipro – 2023

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет: «Комп'ютерні технології і системи»  
Кафедра: «Комп'ютерні інформаційні технології»  
Рівень вищої освіти: бакалавр  
Освітня програма: «Інженерія програмного забезпечення»  
Спеціальність: «121 Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ  
Завідувач кафедри КІТ  
\_\_\_\_\_/Вадим ГОРЯЧКІН/  
(підпис)  
Дата \_\_\_\_\_

### ЗАВДАННЯ

На кваліфікаційну роботу  
бакалавра студенту

1. Тема роботи: «Розробка застосунку «Музична школа»: тренажери музичного слуху»

Керівник роботи: Шаповал Ірина Вікторівна, старший викладач  
затверджені наказом № 77 ст від 07.12.2002.

2. Строк подання студентом роботи: 19.06.2023

3. Вихідні дані до роботи:

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):  
вступ, збір вимог до програмного забезпечення, зовнішнє і внутрішнє  
проектування, тестування та налагодження, висновки, література.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): презентація, відео-демонстрація роботи програми.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи  | Строк виконання етапів роботи | Примітка |
|-------|--|-------------------------------|----------|
| 1     | Постановка задачі  | 03.04.2023 – 05.04.2023       |          |
| 2     | Огляд літератури та аналіз аналогів  | 06.04.2023 – 09.04.2023       |          |
| 3     | Розробка структур вхідних і вихідних даних                                 | 10.04.2023 – 24.04.2023       |          |
| 4     | Визначення вимог до програми.<br>Вибір та обґрунтування мови програмування | 25.04.2023 – 02.05.2023       |          |
| 5     | Узгодження та затвердження ТЗ  | 03.05.2023 – 10.05.2023       | 30%      |
| 6     | Розробка та програмування логіки програми                                  | 11.05.2023 – 18.05.2023       |          |
| 7     | Розробка і реалізація інтерфейсу користувача                               | 19.05.2023 – 26.05.2023       | 60%      |
| 8     | Відлагодження програми   | 27.05.2023 – 02.06.2023       |          |
| 9     | Розробка, узгодження та затвердження програмної документації               | 03.06.2023 – 18.06.2023       | 100%     |
| 10    | Подання кваліфікаційної роботи до кафедри                                  | 19.06.2023 – 26.06.2023       |          |
| 11    | Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії          | 27.06.2023                    |          |

Студент

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (Ім'я ПРІЗВИЩЕ)

Керівник роботи

\_\_\_\_\_ (підпис)

ст. викл. Ірина ШАПОВАЛ

\_\_\_\_\_ (Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка складається з 7 розділів:

- вступ – в даному розділі описується сутність розробки, її актуальність. Складається з 2 сторінки;
- збір вимог до програмного забезпечення – у цьому розділі описуються аналоги програми та література по даній предметній області. Складається з 8 сторінок;
- зовнішнє і внутрішнє проектування – у цьому розділі проведений огляд вхідних і вихідних даних, формалізація задачі, розробка фізичного проекту, приводиться опис об'єктно-орієнтованого проектування, проектування інтерфейсу користувача, ескізи форм, аналіз проекту, проектування динаміки системи, вибір мови програмування. Складається з 23 сторінки;
- тестування та налагодження – включає в себе вибір стратегії тестування, опис тестів методами «чорної» та «білої» скриньки. Складається з 9 сторінок;
- висновки. Складається з 2 сторінок;
- список літератури – включає в себе бібліографічний список використаної літератури. Складається з 1 сторінки;
- додатки – містить робочий проект додаткового без .txt файлу та зображень для форм.

Кількість таблиць: 6 штуки.

Кількість рисунків: 16 штук.

Ключові слова: транспозиція, ступінь ноти, октава.

## ЗМІСТ

|   |    |
|---|----|
| 1. ЗБІР ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....   | 10 |
| 1.1. Перегляд аналогових конкурентів.....         | 10 |
| 1.1.1. Огляд продукту «musictheory» .....         | 10 |
| 1.1.2. Огляд продукту «musicca» .....             | 11 |
| 1.2.1. Головна база теорії музики.....            | 13 |
| 1.2.2. Транспозиція .....                         | 15 |
| Висновки до пункту 1.....                         | 17 |
| 2. ЗОВНІШНЄ І ВНУТРІШНЄ ПРОЕКТУВАННЯ.....         | 18 |
| 2.1. Зовнішнє проектування .....                  | 18 |
| 2.1.1. Функціональне призначення.....             | 18 |
| 2.1.2. Експлуатаційне призначення.....            | 18 |
| 2.1.3. Функціональні вимоги.....                  | 18 |
| 2.1.4. Вхідні та вихідні дані .....               | 19 |
| 2.2. Внутрішнє проектування.....                  | 22 |
| 2.2.1. Аналіз зовнішніх специфікацій систем ..... | 22 |
| 2.2.2. Проектування інтерфейсу користувача .....  | 29 |
| 2.2.3. Проектування динаміки системи.....         | 36 |
| 2.2.4. Вибір мови програмування .....             | 38 |
| Висновки до пункту 2.....                         | 40 |
| 3. ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ.....                | 41 |
| 3.1. Тестування методом «білої скриньки» .....    | 41 |
| 3.2. Тестування методом «чорної скриньки» .....   | 43 |
| Висновки до пункту 3.....                         | 50 |
| ВИСНОВОК.....                                     | 51 |
| ЛІТЕРАТУРА .....                                  | 52 |
| ДОДАТКИ.....                                      | 53 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАК, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ (програмне забезпечення) - є терміном, що використовується для позначення комп'ютерних програм та пов'язаних з ними даних, які забезпечують функціональність та виконання завдань на комп'ютері

Кастомізація – підлаштовування, налаштування чогось під себе. Те ж саме що й персоналізація.

IN (Identify Note) - Спеціальний режим програми де потрібно ідентифікувати ноту, що програється, і вибрати правильний варіант ноти.

TB (Tonality Builder) - режим, де можна візуалізувати вибрані тональності на клавішах піаніно.

Tr (Transposer) - режим зміщення ступені ноти на певну кількість півтонів.

Октава - це музичний інтервал, який включає в себе всі ноти від «до» до «сі», включаючи обидві крайні ноти.

Півтони - музичний інтервал, що дорівнює половині цілого тону.

Ступінь - рух, який відчувається як перехід від одного елемента до іншого, використовуючи зміни у звуковій висоті для вираження процесу руху та розвитку.

## ВСТУП

Музика, як мова виразу та спосіб комунікації, завжди викликала захоплення та захоплювала увагу людей по всьому світу. Вона має безмежну потужність створювати настрої, викликати емоції та зближувати людей, незалежно від їх культурних різниць. Однак, музика - це не лише чудовий світ звуків, але й комплексна система з численними аспектами, які вивчаються у музичній теорії та практиці. Один з таких аспектів - транспозиція, що стає необхідною складовою частиною виконання музики. Транспозер, будь то музичний пристрій або функція, дає можливість змінювати висоту звуків та переносити музичний матеріал в інші тональності або октави. Це надає виконавцям гнучкість і можливість адаптуватися до різних інструментів, голосів або музичних умов. Завдяки транспозеру музиканти можуть зручно грати музику на своїх інструментах без необхідності вивчати нові нотації або перекладати ноти в різні тональності. Самою метою дипломного проекту є створення тренажеру транспозера, який дозволить користувачам покращити свій слух та розуміння музичних понять, зокрема розпізнавання нот та їх октав. Цей тренажер надасть можливість вправлятися в транспозиції і навчитися переносити музичний матеріал в різні тональності та октави. Розроблений тренажер буде ефективним інструментом для музичних студентів, виконавців і навіть просто любителів музики, які бажають покращити свої навички та розуміння музичних концепцій. Він надасть можливість тренуватися у визначенні нот та їх відповідних октав, а також допоможе розширити слухові можливості та розвинути музичну інтуїцію. Результатом роботи буде інтуїтивно зрозумілий і легко використовуваний тренажер, що дозволить користувачам самостійно вдосконалювати свої навички в транспозиції та розпізнаванні нот. Такий тренажер стане цінним доповненням до навчального процесу і допоможе зробити вивчення музики ще більш захоплюючим та результативним. Якщо користувачі не будуть справлятися з транспозицією на слух, то в інтерфейсі будуть точки, які можуть допомогти користувачеві визначити правильну відповідь. Кожен режим повинен мати вільний режим,



де він зможе зробити всі транспозиції і всі тональності для тестів. Інтерфейс повинен мати чорний відтінок, причина такого рішення полягає в тому, що темна тема більше застосовується і у великій кількості попиту. Такі деталі в основному потрібні для комфортності користувача в програмі, потрібні всі умови для приємної практики що вдень. Музика досить велика сфера, де можна зробити дуже багато інструментів для поліпшення своїх базових навичок, але транспозиція рідко має онлайн вигляд або бути в програмному продукті.

Для підтримки нової версії музики або нових стилів потрібно тримати оновлювати та робити все більше нових функцій для написання мелодій. За допомогою такого правила в даний момент люди роблять новий стиль які набрали популярності у світі.

## 1. ЗБІР ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 1.1. Перегляд аналогових конкурентів

«www.musictheory.net» та «www.music». Обидва ресурси спрямовані на зміцнення навичок музикантів, але вони відрізняються відношенням до користувача. Обидва вони мають великий набір тестів і тренажерів, але «www.musictheory.net» пропонує ще більш широкий функціонал у порівнянні з «www.musicca.com».

#### 1.1.1. Огляд продукту «musictheory»

Musictheory – це веб-сайт, який пропонує теоретичні та практичні уроки для визначення нот та інших музичних інструментів з можливістю налаштування режимів. Процес підготовки музиканта полягає у використанні простого інтерфейсу з нотами, де користувач може побачити зображену ноту та вибрати правильну назву для неї. Основна функція режиму «Визначення нот» полягає у виборі правильної ноти з наданого набору варіантів. Користувач може відстежувати кількість вірно визначених нот, але при неправильному виборі кількість спроб збільшується, а відсоток правильних відповідей зменшується з кожним неправильним кліком на варіант (На рисунку 1 наведено приклад інтерфейсу цього режиму).

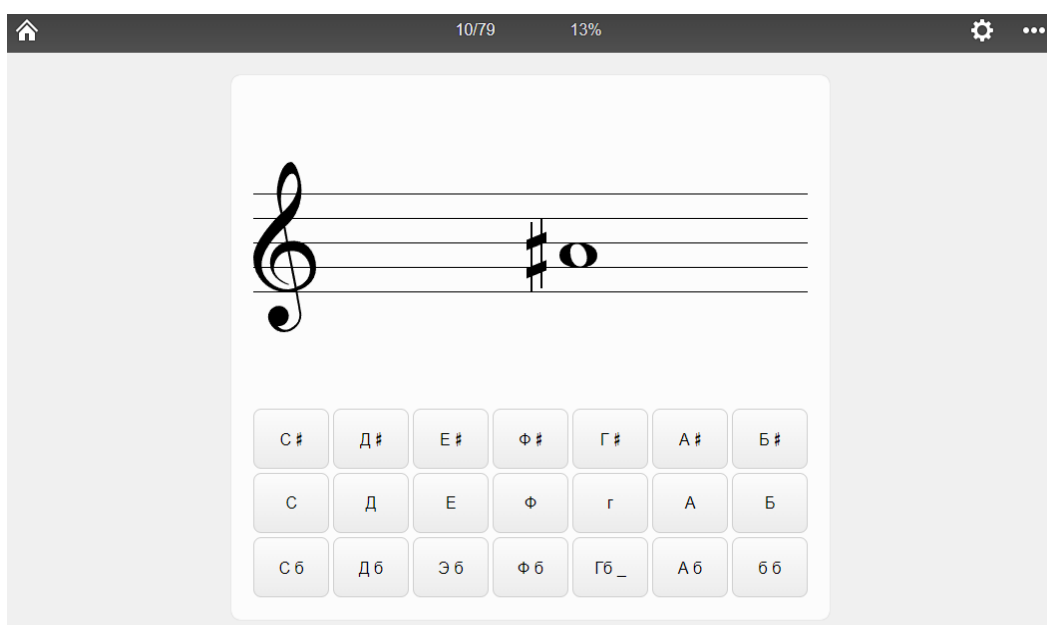


Рисунок 1.1. Приклад інтерфейсу «Musictheory»

#### Переваги:

- Хороший функціонал;
- Кастомізація режимів на смак користувача;
- Анімовані уроки;
- Музичні інструменти (калькулятор);

#### Недоліки:

- Немає регулювання гучності твору ноти;
- Немає маркування клавіш;
- Доволі однотонний дизайн;

Цей веб-сайт має широкий спектр функціоналу, що робить його цікавим для користувачів. Особливою перевагою є можливість налаштування різних параметрів та кастомізація режимів, що дозволяє користувачеві настроїти деталі нот та інші параметри під свої потреби. Однак, основним недоліком сайту є його простий однотонний дизайн без яскравих кольорів, що може бути незахопливим для деяких користувачів. Крім того, кнопка результатів не виділяється яскравим кольором, що може зробити її важкозрозумілою для користувачів. Серед позитивних аспектів сайту варто відзначити наявність анімаційних уроків, які допомагають розуміти музичну теорію, а також наявність вправ з обмеженим вибором інструментів, що збагачують навчальний процес.

#### 1.1.2. Огляд продукту «musicca»

Musicca - це музична платформа, яка пропонує уроки, вправи та інтерактивні інструменти для допомоги музикантам у поглибленому розумінні музики та поліпшенні їх навичок. На платформі користувачі можуть знайти різноманітні матеріали, які допомагають розширити їх музичні знання і навички. Уроки та вправи надають можливість активно вчитися та вдосконалюватися в музиці. Крім того, інтерактивні інструменти дозволяють музикантам експериментувати та створювати музику, надаючи їм практичний

підхід до вивчення. На рисунку 2 показаний приклад інтерфейсу цієї платформи.

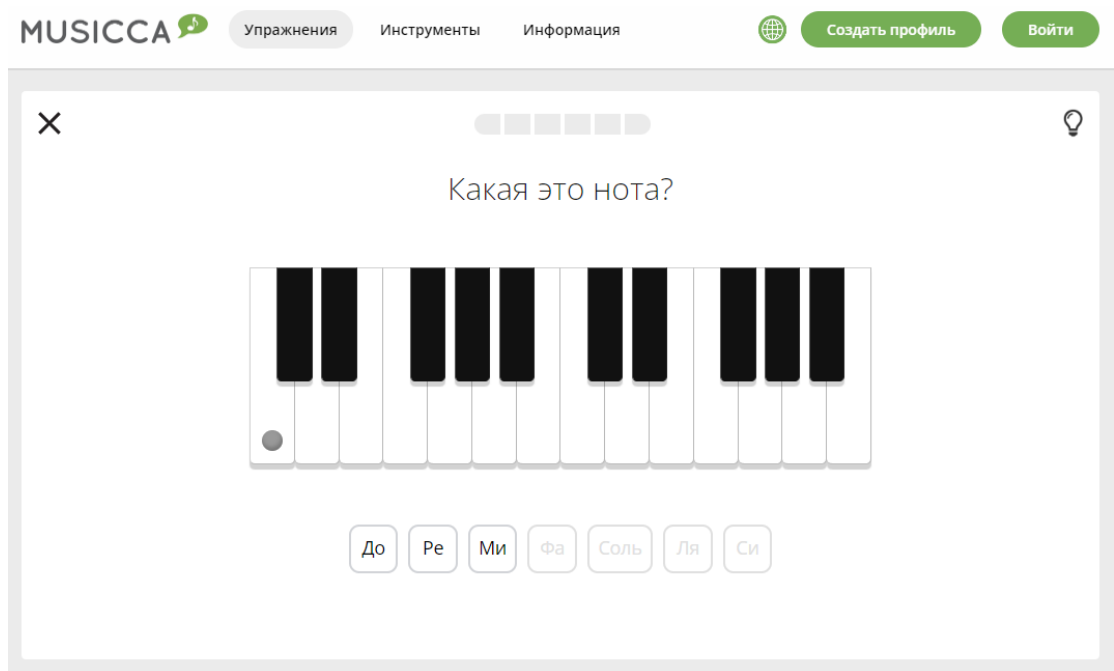


Рисунок 1.2. Приклад інтерфейсу «Musicca»

Переваги:

- Непоганий функціонал;
- Уроки, що використовують частину інтерфейсу прямо на сторінці;
- Музичні інструменти (калькулятор);
- Простий та зручний дизайн;
- Локалізація багатьма мовами;

Недоліки:

- Регулювання гучності нот;
- Відсутня кастомізація режимів;

Однією з приємних особливостей є можливість залишати маркерування на нотах за бажанням користувача, що робить використання сайту дуже зручним. Візуальний дизайн співпадає зі стилістикою піаніно і не надто виділяється, що є перевагою для сайту, оскільки під час роботи деякі яскраві кольори можуть відволікати увагу.

## 1.2. Огляд літератури

### 1.2.1. Головна база теорії музики.

Для досягнення правильного читання і письма необхідно мати знання слів, розуміти їх значення і використовувати граматичні правила, які визначають мову. Аналогічно, музика також має свою теорію, яку треба розуміти, щоб вміти створювати музику, грати на інструменті або просто співати. Теорія музики встановлює правила, які допомагають розбиратися у музичних елементах, таких як ноти, ритм, гармонія і форма. Ці знання і розуміння теорії музики дозволяють нам виразно втілювати свої музичні ідеї і глибше сприймати музичні твори.

Музика складається із нот. Нота – це звук певної висоти. Багато з нас знає назви нот ще зі школи, такі як: «до», «ре», «мі», «фа», «соль», «ля», «сі».

Октава в музиці - це інтервал, в якому частота основного тону змінюється вдвічі. Наприклад, нота в першій октаві має приблизну частоту 27.5 Гц, а в другій октаві - 55 Гц. Склад гармонік цих двох різних звуків має багато спільного, зокрема, це 110 Гц (для третьої октави), 220 Гц (для четвертої октави), 440 Гц (для п'ятої октави) та інші. Тому ноти з однаковими назвами в різних октавах звучать гармонійно, оскільки вони мають спільні вищі гармоніки, які впливають на сприйняття звуку.

Усі сім основних нот, від «до» до «сі», утворюють систему, яку називають октавною системою. Ця система складається з повторюваних нот, які розташовані у послідовності октав. Кожна октава включає сім нот і починається з ноти «до», закінчуючи на ноті «сі». Кожна октава має свою власну висоту звучання, і всередині октави ноти повторюються, проте з різною висотою звучання.

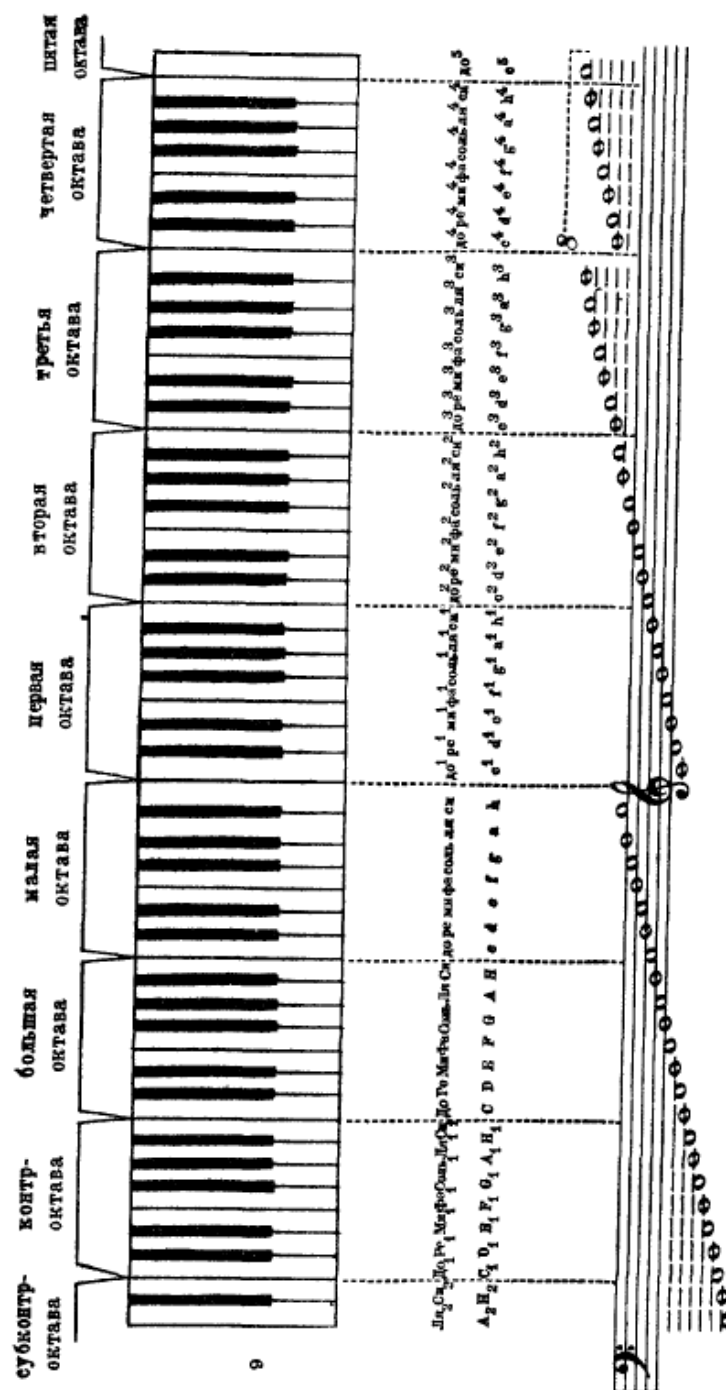


Рисунок 1.3. Загальна октавна система

Наприклад, нота «до» в малій октаві буде звучати нижче, ніж нота «до» в другій октаві. Октавна система, як правило, містить 9 октав. Інші октави, в теорії, можуть існувати, але не використовуються у музиці або є практично непомітними для людського вуха. На рисунку 3 показано список октав в октавній системі, такі як субконтроктава, контроктава, велика октава, мала октава, перша октава, друга октава, третя октава, четверта октава і п'ята октава.

### 1.2.2. Транспозиція

Транспозиція - це переміщення музичного твору, акорду або ноти в іншу тональність на певну кількість інтервалів. Важливо відмітити, що інтервали вимірюються в півтонах, де відстань від «до» до «до діз» складає 1 півтон, а від «до» до «ре» - 2 півтона, що також відповідає 1 тону. Наприклад, від «до» в першій октаві до «до» в другій октаві інтервал складає 12 півтонів, що рівносильно 1 октаві.

Транспозиція використовується як у музиці з тональними системами, так і у музиці без вираженої тоновості. Наприклад, в атональній музиці, такій як додекафонія, транспозиція означає переміщення серії звуків на інший ступінь в межах 12-ти ступенів темперованого звукоряду.

Взагалі кожній кількості інтервалів призначена власна назва, але в рамках доступного функціоналу, що є на даний момент, не будемо детально розглядати різновиди інтервалів. Для розуміння процесу транспозиції достатньо знати, як вимірювати інтервали у півтонах (Картинка 4).

Написано:

Kl. in B

*p dolce*

Звучит:

*p dolce*

Рисунок 1.4. Приклад транспозиції

Основні способи використання транспозиції включають:

1. Пристосування до вокального діапазону: При виконанні вокальних партій, виконавці можуть використовувати транспозицію, щоб змінити висоту партії, щоб вона відповідала їхньому діапазону голосу. Це дозволяє їм зручно співати та виконувати музику без перенапруження голосових зв'язок.

2. Підлаштування до інструментальних можливостей: Деякі інструменти мають обмежений діапазон, особливо у нижніх або верхніх нотах. Використання транспозиції дозволяє виконавцеві грати музичний матеріал в більш комфортних або природних для них висотах, враховуючи обмеження їхнього інструменту.

3. Зміна тонової колористики: Використання транспозиції може змінити відчуття та звучання музичного матеріалу. Наприклад, транспозиція вище може надати яскравості та гостроти звучанню, тоді як транспозиція нижче може створити багатий, глибокий звук.

Для використання транспозиції потрібно знати, на скільки півтонів (або інтервалів) потрібно змінити висоту музичного матеріалу. Це можна зробити, використовуючи музичні терміни, такі як «підняти на октаву» або «знизити на малу терцію», або вказуючи конкретну кількість півтонів для транспозиції. Також важливо мати розуміння відповідних нот та акордів у новому ключі, щоб зберегти правильну гармонію.

Існують різні способи виконання транспозиції, залежно від потреби та використовуваного музичного матеріалу. Деякі музиканти використовують спеціальні нотні листи або програми, які автоматично здійснюють транспозицію для них. Інші можуть ручно переписувати ноти або використовувати транспонуючі інструменти, такі як транспонуючий кларнет або труба.

Незалежно від методу, транспозиція вимагає вміння розуміти музичну теорію, включаючи розміщення нот на нотному стані, інтервали та відносини



між ними. Це дозволить музиканту зручно перенести музичний матеріал у нову висоту та зберегти його музичну структуру та виразність.

Висновки до пункту 1.

Після цієї теоретичної частини можна приділити увагу таким аспектом як те що для транспозера дуже важливо щоб була сама нота (ступінь), вибір октави (є всього три види, small, 1 октава, 2 октава), символ альтерації і півтону. На таких важливих аспектах будуватиметься вся система для транспозиції нот. У свою чергу для користувача буде зроблено тренажер, де він по звуку визначатиме півтони для покращення його слуху до транспонування.

## 2. ЗОВНІШНЄ І ВНУТРІШНЄ ПРОЕКТУВАННЯ

### 2.1. Зовнішнє проектування

#### 2.1.1. Функціональне призначення

Функціональним призначенням є створення тренажерного режиму для транспозиції. Користувач може тренуватись у вільному режимі (TOOL) та вчитися користуватися цим інструментом для розширення своїх базових знань. Також можливе програвання нот на слух та визначення правильної відповіді для користувача. Крім того, програма дозволяє переносити ступені та акорди на задану кількість півтонів (Transposer) і має режим навчання.

#### 2.1.2. Експлуатаційне призначення

Експлуатаційне призначення програми:

- збільшити інтерес до інструменту, як Transposer;
- зручне зміщення висоти музичних нот чи акордів на задану кількість півтонів чи інтервалів;
- тренажер для музикантів, що працює з інструментом як Transposer;
- вільний інструмент, де користувач може спробувати різні звукові ноти на слух або побачити їх на екрані для ідифікації правильної відповіді в режимі Transposer (LEARN);

#### 2.1.3. Функціональні вимоги

Музична програма має виконувати такі види вимог як:

- почати тренування для Transposer;
- вимикати або регулювати звук для режиму;
- малювати модель звуку на октаві, який чує користувач;
- програвати звук нот;
- унікальний інструмент, де він може спробувати Transposer;
- регулювання звуку користувача;

## 2.1.4. Вхідні та вихідні дані

### 2.1.4.1. Вхідні дані

У режимі навчання виходять такі вхідні, як:

1) `isJustBegan` - true/false значення, що позначає чи генерація виконується вперше. У випадку, якщо це вперше (true) - виконується початкова ініціалізація внутрішніх даних класу транспонувальника: `correctAnswers_`, `transpositionsGenerated_`, `generatedRange_`.

2) `semitones` - int значення (ціле число), максимального значення не має, мінімальне - від 0. Позначає кількість півтонів між нотами, у якості відповіді під час тренувального випробування, де на екрані відображено ці дві ноти.

У режимі інструменту виходять такі вхідні, як:

3) `note` - значення структури `Note`. Структура `Note` вміщує в собі дані: `degree_` - позначає ступінь ноти типу `enum Degree` (C, D, E, F, G, A, B) , `alterationSign_` - позначає знак альтерації типу `enum AlterationSign` ( SHARP, FLAT, NONE, BEKAR, DOUBLE\_SHARP, DOUBLE\_FLAT ), `octave_` - позначає октаву типу `enum Octave` (SUB\_CONTRA, CONTRA, GREAT, SMALL, LINE1, LINE2, LINE3, LINE4, LINE5, LINE6). Ця змінна описує ноту, яку потрібно транспонувати інструментом транспонування.

4) `semitones` - int значення (ціле число), максимальне значення 12, мінімальне -12. Позначає кількість півтонів, на яку потрібно транспонувати ноту `note` інструментом транспонування.

### 2.1.4.2. Вихідні дані

У режимі навчання виходять такі вихідні, як:

1) `correctAnswers_` - int значення (ціле число), максимального значення не має, мінімальне - від 0. Позначає кількість вірних відповідей, наданих користувачем під час тренувальних випробувань з визначення кількості півтонів.

2) `transpositionsGenerated_` - int значення (ціле число), максимального значення не має, мінімальне - від 0. Позначає кількість виконаних генерацій транспозицій для тренувальних випробувань для користувача.

У режимі інструменту виходять такі вихідні, як:

3) correctAnswers\_ - int значення (ціле число), максимального значення не має, мінімальне - від 0. Позначає кількість вірних відповідей, наданих користувачем під час тренувальних випробувань з визначення кількості півтонів.

4) transpositionsGenerated\_ - int значення (ціле число), максимального значення не має, мінімальне - від 0. Позначає кількість виконаних генерацій транспозицій для тренувальних випробувань для користувача. - выходные: 1) resultNote - значення структури Note (описана вище). Позначає результуючу ноту, що утворюється після виконання операції транспонування інструментом транспонування.

### 2.1.5. Опис зовнішнього інформаційного середовища

Для функціонування даної програми потрібна операційна система Windows 7, а також завдяки фреймворку Qt програма може підтримувати і Linux. Також для коректної роботи системи будуть потрібні певні бібліотеки від Qt версії 6.4.0 та новіших. Також для користувача потрібні динаміки з можливістю програвати звук у самій програмі, або навушники, які передають звук від програми. Мікрофон для програми не потрібен і не використовуватиметься.

Параметр, який дозволяє користувачу регулювати гучність нот під час вправ у режимі. Також присутня кнопка, яка відтворюватиме ноту до тих пір, поки користувач не завершить вправу. Доступно декілька режимів транспозиції для вибору користувачем. Додатково, присутній вільний режим інструменту, де користувач може тренуватися з транспозицією, встановлюючи параметри, такі як нота, знак альтерації, октава, півтон. Показ результатів та правильних відповідей в активних верхніх вікнах та у кінцевому фініші тренування показ результатів.

Користувач має можливість керувати процесом транспозиції, у будь-який момент він може зупинити ноти, які програватимуться після транспозиції та давати параметри для транспозиції. У трьох режимах є вільні режими, де користувач може спробувати абсолютно всі налаштування і прослухати всі варіанти, які можуть бути в самому режимі. Вхідними даними керує також користувач у режимі вивчення та практики.

Функціональні вимоги в специфікації викладені у формі прецедентних діаграм (Рисунок 5.)

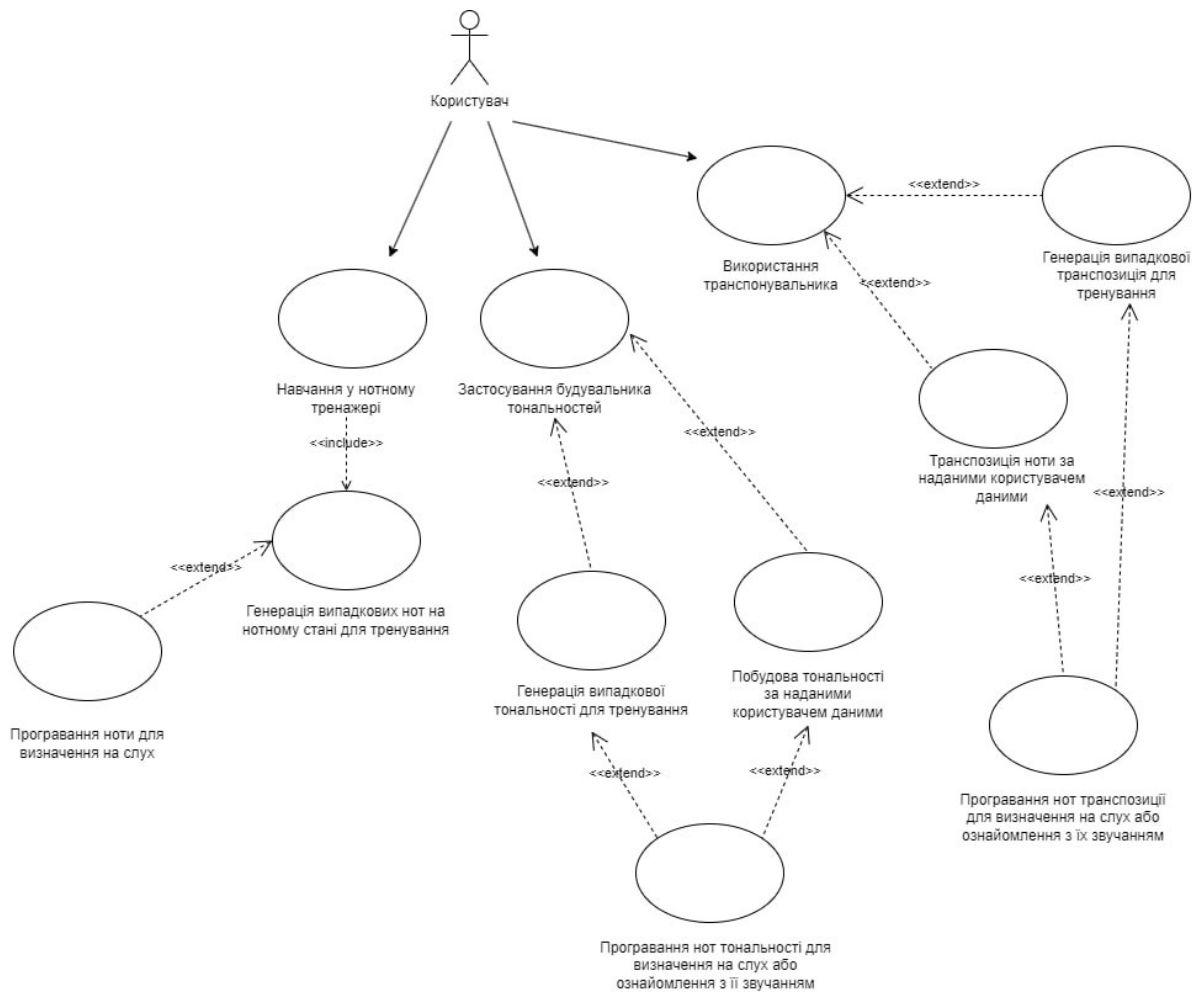


Рисунок 2.1. Діаграма прецедентів

## 2.2. Внутрішнє проектування

### 2.2.1. Аналіз зовнішніх специфікацій систем

#### 2.2.1.1. Моделювання словника системи

Користувач має можливість взаємодій з такими сутностями як: MainWindow (Головний клас який приймає методи та всі функціональні функції для взаємодії з об'єктами які знаходяться в самому класі), TransposerController (методи де ініціалізуються старт роботи інструментів), Transposer (генерує випадкову транспозицію двох нот для тестів режимі вправи, які відповідають за функціональність самого транспозера режиму).

1. MainWindow - Головна частина всього інтерфейсу для користувача, має безліч об'єктів, де реалізовані в TransposerController.
2. TransposerController - Керує частиною процесу старту і самої вправи у тому числі приймає такі дані від користувача як ступінь, октаву,

напівтону та знак альтерації. Керує процесами розтавання крапок на фортепіано.

3. Transposer - Частина відповідального за генерацію та обробку правильних та неправильних відповідей користувача під час дії програми.

Таблиця 2.1. – Сутності, атрибути та методи

| Сутність   | Атрибути | Методи   |
|------------|----------|--|
| 1          | 2        | 3  |
| MainWindow | –        | <p>on_aboutUsButton_clicked – Відкриває сторінку About Us інформацію.</p> <p>on_semitonesToTranspose_Learn_currentIndexChanged – Перевіряє чи вибрав користувач півтони, щоб розблокувати перехід до наступної транспозиції в режимі навчання.</p> <p>on_learnButtonT_clicked – Виконання дії розпочати вправу.</p> <p>on_toolButtonT_clicked – Виконання дії розпочати вправи у вільному режимі.</p> <p>on_resultPageExitButton_3_clicked –</p> <p>on_TransposerButton_clicked –</p> <p>on_semitonesToTranspose_currentIndexChanged – Перевіряє чи вибрав користувач півтони для розблокування транспозиції.</p> <p>on_octaveToTranspose_currentIndexChanged – Перевіряє чи вибрав користувач октаву щоб розблокувати транспозицію.</p> |

Продовження таблиці 2.1

|                      |  |   |
|----------------------|--|---|
|                      |  | <p>on_signToTranspose_currentIndexChanged – Перевіряє чи вибрав користувач символ альтерації для розблокування транспозиції.</p> <p>on_noteToTranspose_currentIndexChanged – Перевіряє чи вибрав користувач ступінь альтерації для розблокування транспозиції.</p> <p>on_exitButton_3_clicked – Виконує дію виходу з режиму на початкову сторінку режиму.</p> <p>on_playTP_clicked – Відтворювати побудовану користувачем звук із нотами.</p> <p>on_transposeButtonTP_clicked – Виконує транспозицію за заданими параметрами користувача.</p> |
| TransposerController | <p>currentTransposition_ – Поточна побудована транспозиція.</p> <p>isEndless_ – Виконує роль того чи нескінченним режим чи ні.</p> <p>isToolMode_ – Виконує роль</p> | <p>BeginLearnFlow – Ініціалізує початок роботи тренажеру.</p> <p>BeginToolFlow – Ініціалізує початок роботи інструментарію.</p> <p>ControlTransposeNextButtonState – Активує/деактивує кнопку Transpose/Next, зокрема. змінюючи дизайн.</p> <p>IsToolMode – Повертає режим.</p> <p>ProceedAnswer – Обробляє відповідь, дану користувачем.</p>   |



## Продовження таблиці 2.1

|            |  |  |
|------------|--|--|
|            | <p>того, чи є в режимі інструменту в даний момент.</p> <p>maxAmount_ – Вибрана максимальна кількість тональностей.</p>   | <p>Transpose – Виконує транспозицію.</p> <p>PlayTransposition – Відтворює поточну тональність.</p> <p>ControlPlayButtonState – Активує/деактивує кнопку Play, у т.ч. змінюючи дизайн.</p> <p>SetTranspositionDotsOnKeys – Встановлює позиції точок відповідно до транспозиції.</p> <p>HideDotsOnKeys – Приховує крапки на клавішах піаніно, коли йде наступна транспозиція.</p> <p>HideShowLearningModeElements – Приховує/показує елементи навчання процесу LEARN.</p> <p>SelectedIndexToTranspositionsAmount – Зчитує значення максимальної кількості тональностей для вгадування.</p> |
| Transposer | <p>correctAnswers_ – Кількість чітко вгаданих транспозицій.</p> <p>generatedRange_ – Поточний згенерований інтервал для транспозиції для режиму LEARN користувача.</p> | <p>GenerateTransposition – Генерує випадкову транспозицію двох нот.</p> <p>IsCorrectAnswer – Обробляє відповідь користувача по транспозиції (тільки для півтони в режимі LEARN).</p> <p>TransposeNote – Транспонує вхідну ноту на задану кількість напівтонів.</p>   |

## Закінчення таблиці 2.1

|  |  |  |
|--|--|--|
|  | transpositionsGenerated_<br>– Поточна кількість<br>згенерованих<br>транспозицій. | GetCorrectAnswers – Повертає<br>кількість вірних відповідей.<br>GetTranspositionsGenerated –<br>Заперечує кількість генерації<br>програми для старту навчання<br>користувача.<br>NullifyData – Обнуляє дані щодо<br>вгадування для транспозиції. |
|--|--|--|

## 2.2.1.2. Моделювання розподілу обов'язків у системі

Багато класів які працюють разом для отримання певних результатів для користувача показано на таблиці 2.2.

– TransposerController, Transposer – передача створеної тональності на екран користувача. Також обробка відповідей у режимі LEARN тональності для користувача на LIVE-Борді і наприкінці вправи. Також робота звичайного інструменту TOOL.

– MainWindow, TransposerController – передача дій з об'єктами яка повинна відбуватися залежно від того, що вибере користувач в інструменті транспозера.

Таблиця 2.2. – Моделювання залежностей

| Клас,<br>який зав'язується | Клас, з якого<br>зав'язується | Тип зв'язку |
|----------------------------|-------------------------------|-------------|
| 1                          | 2                             | 3           |
| TransposerController       | Transposer                    | Реалізація  |
| Transposer                 | -                             |             |
| MainWindow                 | TransposerController          | Агрегація   |

## 2.2.1.3. Визначення призначень об'єктів за допомогою CRC карток

Можна використовувати CRC-картки (Class - Responsibility - Collaboration) для конкретизації класів.

У таблицях 2.3. – 2.5. наведено CRC-картки на класи проекту.

Таблиця 2.3. – CRC-картка для класу MainWindow

| Базовий клас  | Похідні класи (нащадки) |
|---|-------------------------|
| відсутні  | відсутні                |
| Обов'язки   | Зв'язки                 |
| Головне вікно, де будуть посилатися всі інші класи для виконання певних методів з об'єктами | TransposerController    |

Таблиця 2.4. – CRC-картка для класу TransposerController

| Базовий клас  | Похідні класи (нащадки) |
|---|-------------------------|
| відсутні  | відсутні                |
| Обов'язки   | Зв'язки                 |
| <ol style="list-style-type: none"> <li>Ініціалізує початок роботи тренажера.</li> <li>Ініціалізує початок роботи інструментарію.</li> <li>Зактивує/деактивує кнопку Transpose/Next, у т.ч. змінюючи дизайн.</li> <li>Повертає режим.</li> <li>Обробляє відповідь, даний користувачем.</li> <li>Виконує транспозицію.</li> <li>Програє поточну тональність.</li> <li>Активує/деактивує кнопку Play, в т.ч. змінюючи дизайн.</li> <li>Встановлює позиції точок відповідно до транспозиції.</li> </ol> | Transposer              |

## Продовження таблиці 2.4

|   |  |
|---|--|
| 10. Ховає точки на клавішах піаніно.                                    |  |
| 11. Ховає/показує елементи для навчання.                                |  |
| 12. Зчитує значення максимальної кількості тональностей для вгадування. |  |

Таблиця 2.5. – CRC-картка для класу Transposer

| Базовий клас  | Похідні класи (нащадки) |
|---|-------------------------|
| відсутні  | відсутні                |
| Обов'язки   | Зв'язки                 |
| 1. Генерує випадкову транспозицію двох нот.<br>2. Обробляє відповідь користувача по транспозиції (у напівтонів).<br>3. Транспонує вхідну ноту на задану кількість напівтонів.<br>4. Повертає кількість вірних відповідей.<br>5. Повертає кількість генерацій.<br>6. Обнуляє дані по вгадуванню. | відсутні                |

## 2.2.1.4. Побудова об'єктної моделі (діаграми класів)

Клас `MainWindow` має композиційний зв'язок з класом `TransposerController`, тоді як `TransposerController` має залежність від класу `Transposer`. Це означає, що об'єкт класу `TransposerController` є частиною об'єкту класу `MainWindow`, і використовує функціональність класу `Transposer` для виконання певних завдань. Така структура допомагає організувати і взаємодіяти між різними

компонентами програми, забезпечуючи їх взаємодію та функціональність.  
(Рисунок 6)

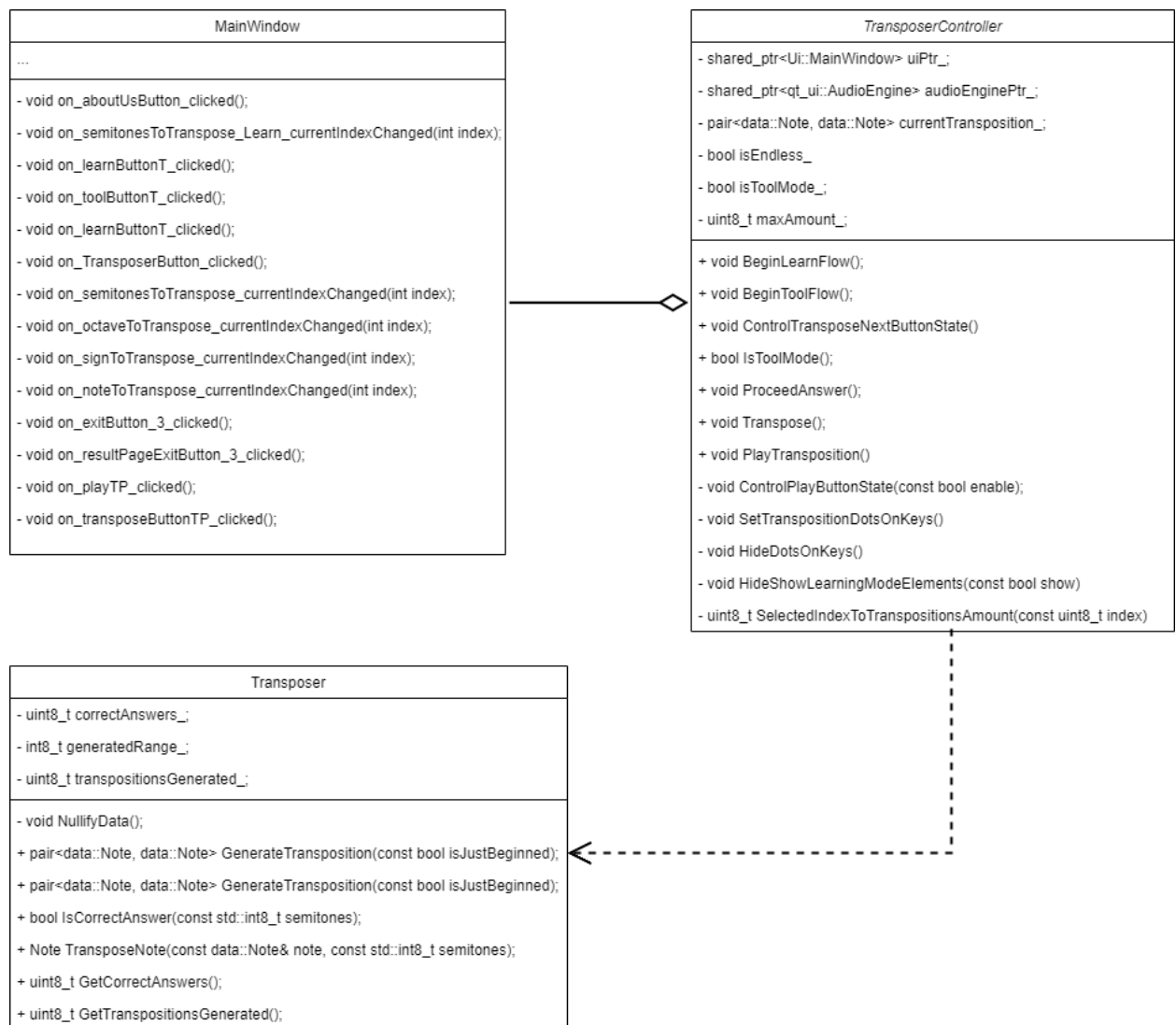


Рисунок 2.2. Діаграма класів

## 2.2.2. Проектування інтерфейсу користувача

### 2.2.2.1. Створення ескізів форм

Для створення відповідного інтерфейсу для користувача при якому всі компактні і потрібні деталі можна виділити такі аспекти як:

меню переходів з режиму на інший режим;

в режимі вибору параметрів для режиму (В даному випадку вибір кількість нот);

відображення вибраних нами тональностей для побудови композиція, яку зробив користувач;

музичний супровід під час вправи в режимі;

показувати кількість спроб та кількість правильних відповідей на музичні вправи;

віконце для керування звуку для користувача;

піаніно де буде з'являтися точки вибрані заздалегідь налаштувань користувача;

Користувач повинен мати швидкий доступ до бокового меню який включає такі кнопки як:

«Home» - Головна сторінка де буде показана інформація про програму та лого програми;

«Identify Note» - Перший режим Elegia де потрібно ідентифікувати ноти за звучанням та нотами, які будуть показуватися на екрані користувача;

«Tonality Builder» - режим, де можна візуалізувати вибрані тональності на клавішах піаніно;

«Transposer» - режим зміщення ступені ноти на певну кількість півтонів. Містить два режими - режим навчання Learn та режим інструменту Tool.

З такими вимогами для користувача будуть зручними такі види вікон:

головне меню (Рисунок 2.3);

перший режим Identify Note;

меню тренажера IN (Рисунок 2.4);

тренажер IN (Рисунок 2.5);

другий режим Tonality Builder;

меню тренажера ТВ (Рисунок 2.6);

тренажер ТВ (Рисунок 2.7);

третій режим Transposer;

меню тренажера Tr (Рисунок 2.8);

тренажер Tr (Рисунок 2.9);

Головним інструментом є головна панель збоку програми. Там надано три режими для тренування з музикою. Кожен режим має свою початкову частину де описується сам режим і представлений вибір налаштування спроб для користувача, у тому числі користувач може вибрати нескінченні спроби і тренуватися до тих пір, поки не зробить певну кількість помилок. Взаємодія користувача з програмою може лише через комп'ютерну мишу. Користувачеві потрібно тільки перемикати режими музики і також вибирати налаштування режиму і користуватися бічною панеллю для швидкого доступу. IdentifyNote має інтерфейс нотного стану для користувача, суть роботи полягає в тому, що виресовуються ноти на нотному стані і користувачеві потрібно буде по зображенню зрозуміти що за ноти. Залежно від того, де буде ця нота, він зможе її індефікувати.

TonalityBuilder має інтерфейс фортопіано, де будуть вимальовуватися точки для допомоги в індифікації тональності. Також буде присутня кнопка для звуку знегенерованої тональності, користувач зможе індифікувати і по слуху тональність якщо це в його силах.

Transposer має інтерфейс фортепіано та має точки для показу користувачеві транспозиції. В інтерфейсі буде кнопка PLAY де він може прослухати транспозицію і вибрати правильну відповідь, якщо у нього є досвід. Крапки на фортепіано можуть допомогти йому індефікувати правильну відповідь, якщо буде працювати з теорією.



Рисунок 2.3. Ескіз начального вікна Home класів

На малюнку 2.3 показана головна частина сторінки Home, це стартова сторінка де користувач читатиме опис програми та попередження. Також користувачеві представимо наше лого і додамо картинку щоб було виділити написаний текст.

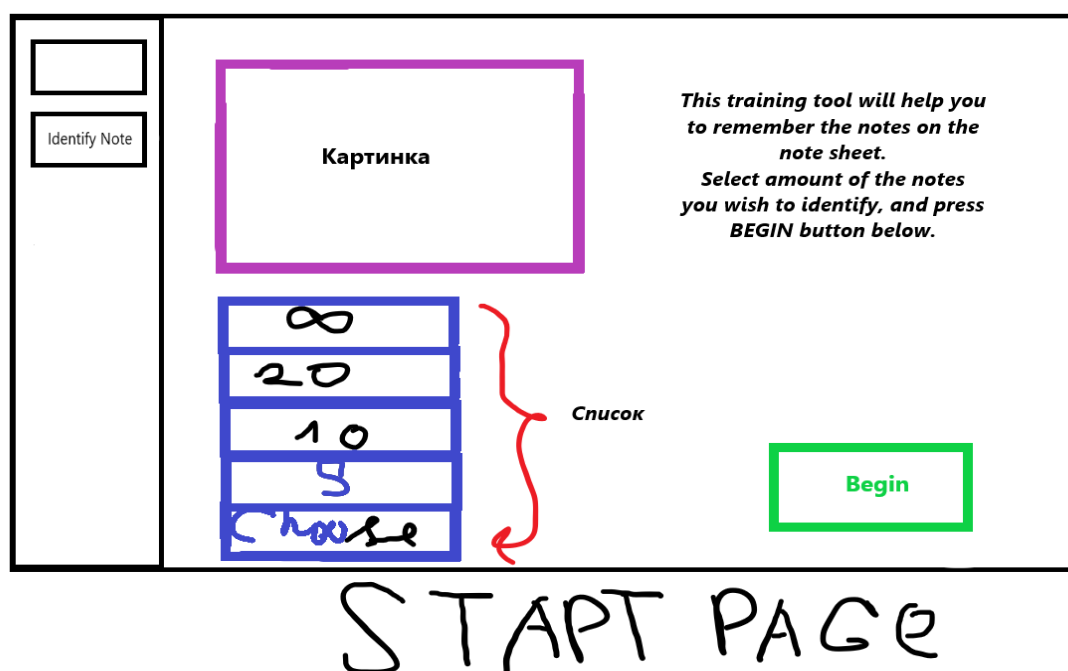


Рисунок 2.4. Ескіз начального вікна Identify Note



На малюнку 2.4 ескіз початкового вікна IN де користувач вибирає кількість спроб зі списку та активує режим на кнопку Begin.

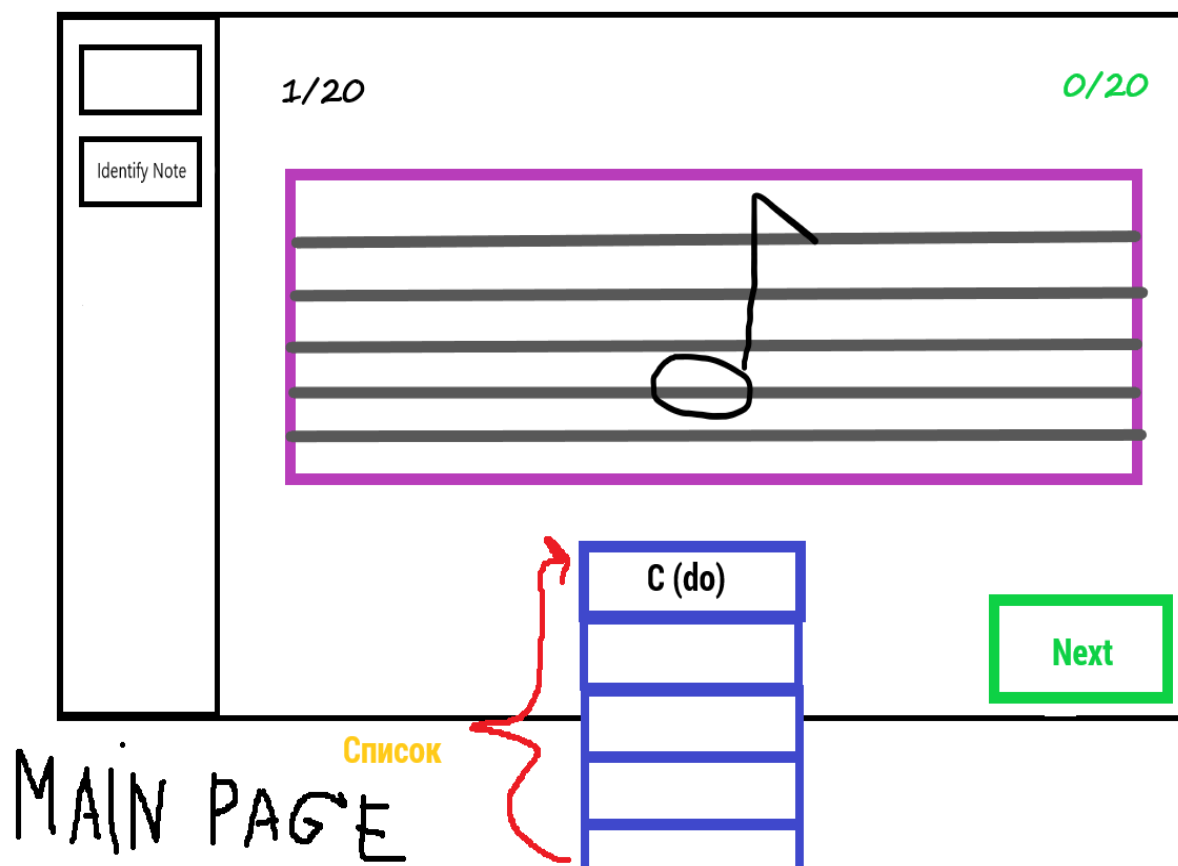
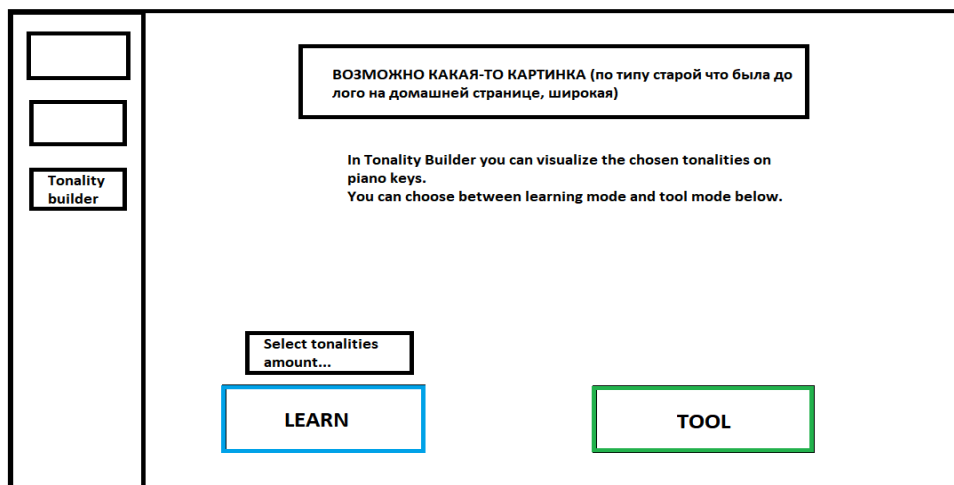


Рисунок 2.5. Ескіз головного вікна Identify Note

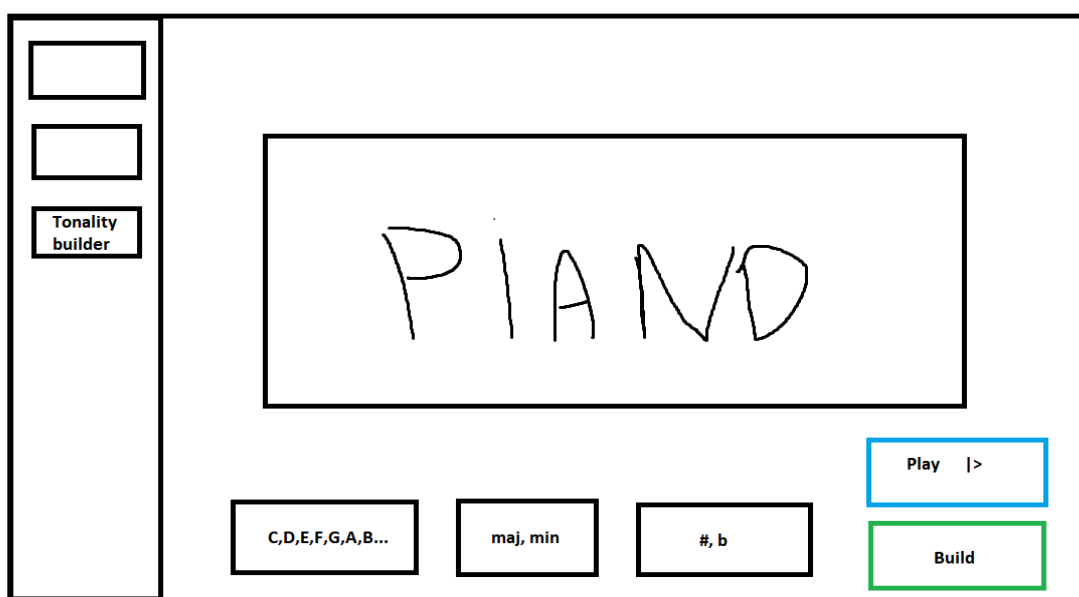
На малюнку 2.5 показаний ескіз головного вікна IN, де користувач буде дивитися на робочу програму. На малюнку зображено фото корда де вимальовуватиметься нота, за становищем на цьому корді ноти користувач повинен вибрати один з варіантів відповідей ноту і натиснути Next для отримання бала вправо верхнє число. При неправильній спробі кількість спроб скорочуватиметься щоразу. Інтерфейс повинен мати однотонні кольори але потрібно виділити відповіді яскравішим кольором щоб дати зрозуміти користувачеві де відразу знаходитися пункт з відповідями.



START PAGE

Рисунок 2.6. Ескіз начального вікна Tonalities Builder

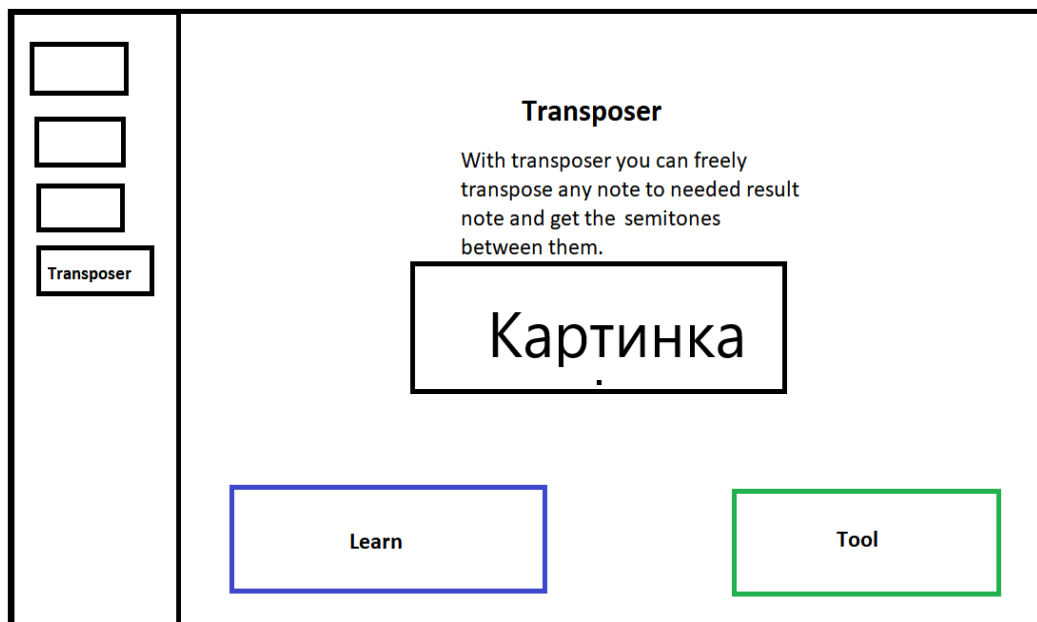
На малюнку 2.6 показу ескіз початкового екрана вибору користувачем налаштувань для ТВ. На сторінці дві кнопки, де одна відповідає за навчання, а друга за запуск цього режиму. Також можна вибрати кількість тональностей для користувачів. Усього тональностей від п'яти до нескінченності, залежно від цього, у користувача буде стільки кількість проб.



MAIN PAGE

Рисунок 2.7. Ескіз головного вікна Tonalities Builder

На малюнку 2.7 показаний ескіз головного вікна тренажера ТВ, де користувач може вибрати під свої налаштування тональності і програти їх на кнопку Play.



START PAGE

Рисунок 2.8. Ескіз начального вікна Transposer

На малюнку 2.8 показаний ескіз початкового вікна Transposer, де є дві кнопки. Одна навчає, як користуватися даним тулом, друга є вже використанням даного інструменту з метою користувача. Також на сторінці була додана картинка з текстом де описує даний режим і картинка виділяє текст і має приємну картинку для користувача. Інтерфейс буде мати однотонний темний тон, щоб не було навантаження на очі для користувачів. Будь-якого дня вона буде чорна і місцями різнокольорова. Текст також буде однотонний колір який буде підходити для зображення з іншими режимами як IdentifyNote і TonalityBuilder. На цій сторінці буде вибір кількості згенерованих транспозицій для відгадування користувачем.

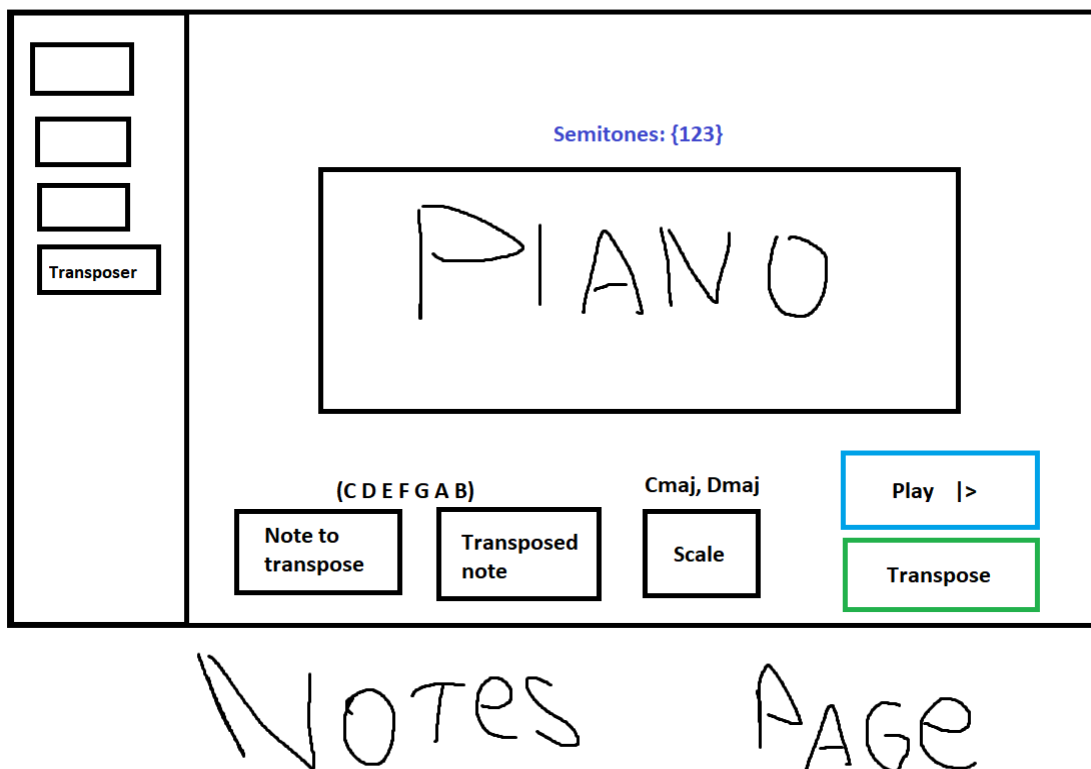


Рисунок 2.9. Ескіз головного вікна Transposer

На малюнку 2.9 показаний ескіз головного вікна Transposer. У робочому стані дана сторінка повинна бути з піаніно, де відображається точки з заздалегідь вибраними налаштуваннями користувачем. При натисканні кнопки play відіграватиметься звук із вибраними нотами. Кнопка Transpose – режим зміщення ступені ноти на певну кількість півтонів.

### 2.2.3. Проектування динаміки системи

Користувач керує всією системою, яка йде в процесі використання програми. Основне завдання користувачів є введення даних через подані підготовлені дані про музику. Виходячи з цієї інформації, яка у нас є, можна побудувати такі сценарії послідовності як:

- керування параметрами IdentifyNote (Рис. 2.2.1);
- керування параметрами TonalityBuilder (Рис. 2.2.2);
- керування параметрами Transposer (Рис. 2.2.3);

Всі сценарії мають загальну структуру послідовності через загальну одну мету - дати користувачеві більше досвіду у сфері музики.

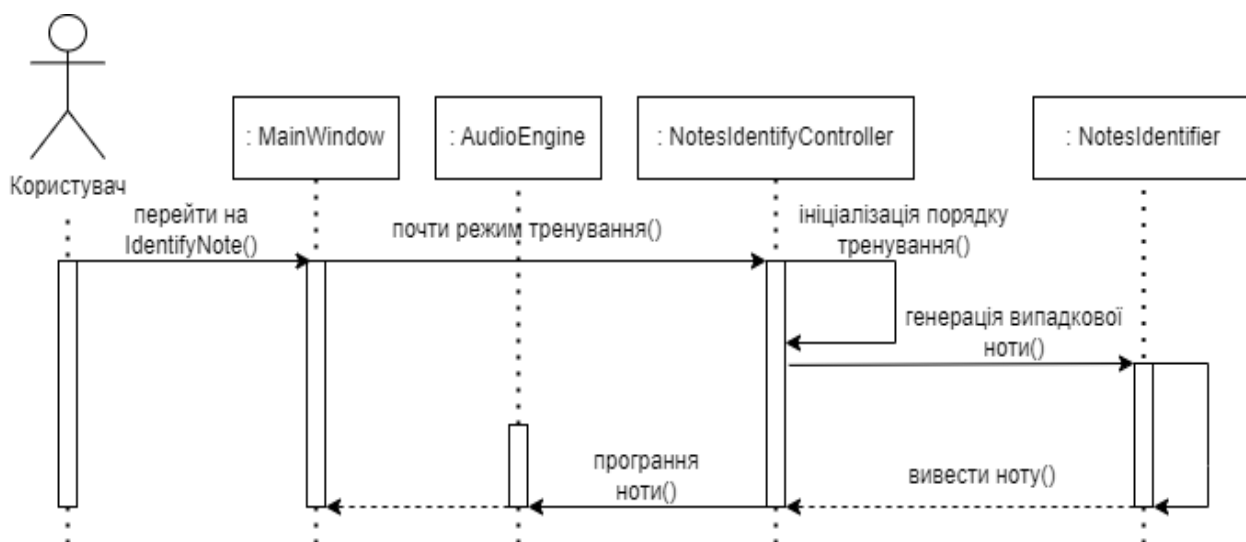


Рисунок 2.2.1. Діаграма послідовності для варіанту використання «IdentifierNote»

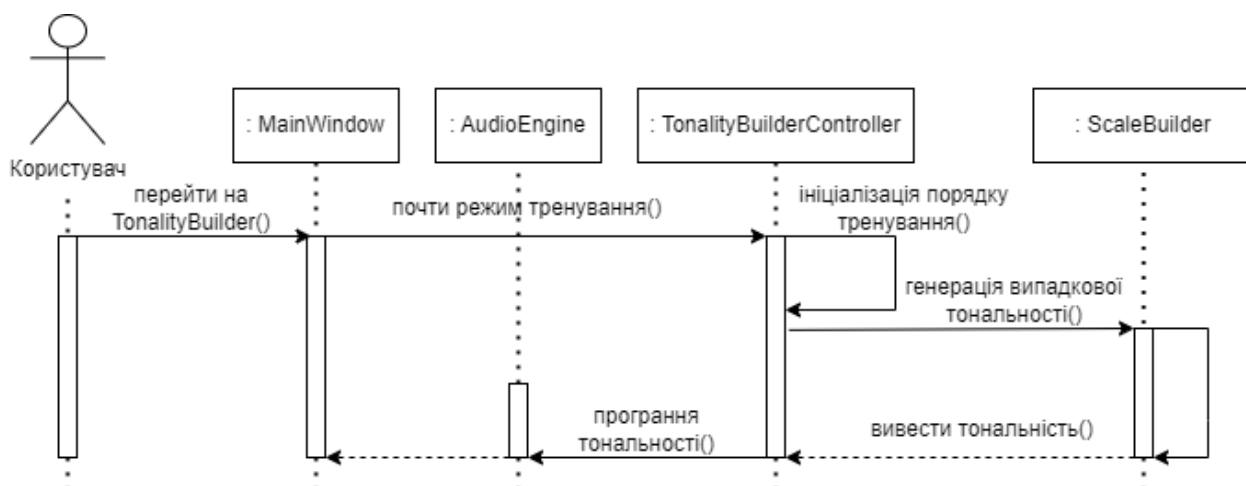


Рисунок 2.2.2. Діаграма послідовності для варіанту використання «TonalityBuilder»

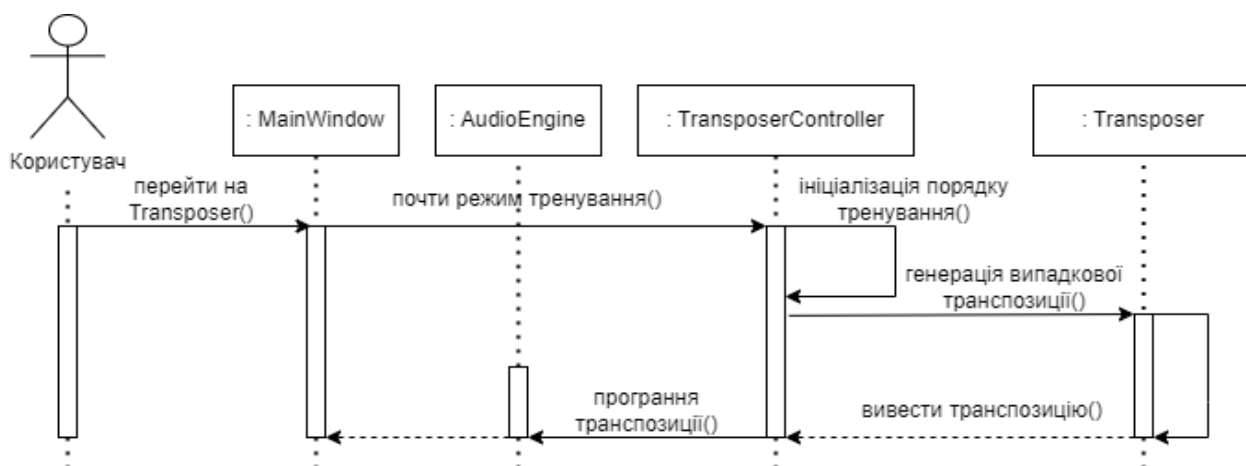


Рисунок 2.2.3. Діаграма послідовності для варіанту використання «Transposer»

#### 2.2.4. Вибір мови програмування

Для розробки цієї програми було обрано окремий фрейворк QT. У такому фреймворку є багато плюсів, включаючи, що підтримка кроссплатформенного програмного забезпечення на основі C++. Також наявністю багато бібліотек, наприклад:

- Python — PyQt, PySide;
- Ruby — QtRuby;
- Java — QtJambi;
- PHP — PHP-Qt;

Головний аспект Qt є Qt Creator, де створення інтерфейсів обходиться набагато простіше, ніж в інших середовищах розробки. У Qt існує система сигналів-слотів. Це така система, де сигнал – це умовний тригер, який посилає сигнал у пов'язаних слотах. Тобто це можуть бути: натискання на кнопки, зміна положення перемикачів (радіо-кнопок та інше). В ролі слотів - це обробники таких сигналів. Тобто те, що треба робити, якщо надходить такий-то сигнал.

Це дуже допомагає робити програму динамічнішою - наприклад при натисканні на кнопки, змінювати їх дизайн. Або можна ще змінювати дизайн при наведенні на кнопки.

Стили кнопок та інтерфейсу робляться майже як у CSS. QT має власну інтерпретовану мову QSS, що схожа на CSS. Також QT має систему ресурсів, у які можна завантажувати файли зображень, аудіо и тд, які можна застосовувати у програмі.

Qt не лише включає в себе бібліотеку та модулі, але також має додаткове програмне забезпечення та утиліти:

Qt Creator: Qt Creator є інтегрованою середовищем розробки (IDE), що дозволяє писати, компілювати та запускати код, а також виконувати тестування та налагодження програм. Це програмне середовище доступне для операційних систем Windows, Linux і macOS.

Qt Designer: Цей інструмент дозволяє ефективно створювати графічні користувацькі інтерфейси (GUI). Він входить у комплекті з фреймворком і призначений для розробки програм, де візуальні компоненти відіграють важливу роль. Створення інтерфейсу відбувається в середовищі за допомогою мови програмування C++, після чого зберігається у файлі і підключається до проекту, розробленого з використанням Qt.

Для програми Elegia був вибраний спеціальний Cmake (кросс-платформовий відкритий генератор сценаріїв складання). CMake не виконує безпосередньо процес складання, але його головна функція полягає у створенні файлів управління складанням на основі файлів CMakeLists.txt.

З головних плюсів із Cmake можна виділити такі аспекти як:

- проста мова сценаріїв;
- засоби розширення функціональності через модулі;
- мінімальне число залежностей;
- підтримка кешування;
- наявність засобів для крос-компіляції;
- підтримка генерації файлів складання для широкого спектра систем складання і компіляторів;
- наявність утиліт ctest і cpack для визначення сценаріїв тестування і складання пакунків;
- утиліта cmake-gui для інтерактивного налаштування параметрів складання;

Початковий код самого CMake був написаний мовою C++, з ліцензією BSD (Ця ліцензійна угода вперше застосовувалась для поширення операційних систем, подібних до Unix, з роду BSD).

З самого початку проекту були вибрані спеціальні налаштування CMake для CMakeLists для зручної розробки та запуску.

Висновки до пункту 2.

На стадії проектування системи була збільшена увага в такій області, як ескізи інтерфейсу для користувача. Мати однотонний приємний інтерфейс є першим аспектом, на що дивиться відразу користувач. Було також проектування діаграм для програми. Було обрано мову програмування та розписані плюси щодо цієї мови.

Хочу звернути увагу щодо CMake, генератор сценарією допоміг у збиранні із самим QT та в інших ситуаціях. Є складність у тому, що для спільної розробки потрібно налаштовувати та встановлювати версії Msys64 і робити повне налаштування QT. З самого початку проекту були вибрані спеціальні налаштування CMake для CMakeLists для зручної розробки та запуску.



### 3. ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ

#### 3.1. Тестування методом «білої скриньки»

Опис: транспозиція ноти на заданий інтервал. Коли: транспонується нота  
Тоді: функція повертає правильну ноту, що транспонується.

```
TEST(ElegiaTransposerTest, TransposeNote)
{
    data::Note note{ data::Degree::C, data::AlterationSign::NONE,
data::Octave::LINE1 };
    std::int8_t transposeRange1{ 14 };
    std::int8_t transposeRange2{ -14 };
    data::Note expectedNote1{ data::Degree::D, data::AlterationSign::NONE,
data::Octave::LINE2 };
    data::Note expectedNote2{ data::Degree::B, data::AlterationSign::FLAT,
data::Octave::GREAT };
    ASSERT_EQ(elegia::Transposer::TransposeNote(note, transposeRange1),
expectedNote1);
    ASSERT_EQ(elegia::Transposer::TransposeNote(note, transposeRange2),
expectedNote2);
}
```

Опис: генерація транспозиції та її вгадування з правильною відповіддю  
Коли: згенерований інтервал збігається із запропонованою користувачем  
відповіддю Тоді: функція перевірки повертає true

```
TEST(ElegiaTransposerTest, CorrectRange)
{
    // Повертає пару нот.
    const auto generatedTransposition{
elegia::Transposer::GenerateTransposition(true) };

    data::Note currentNote{ generatedTransposition.first };
```

```

std::int8_t semitones = 0;
// Якщо друга нота вища за першу - інтервал позитивний.
if (generatedTransposition.first < generatedTransposition.second)
{
    do
    {
        ++currentNote;
        ++semitones;
    } while (!(currentNote == generatedTransposition.second));
}
// Якщо друга нота нижче за першу - інтервал негативний.
else if (generatedTransposition.first > generatedTransposition.second)
{
    do
    {
        --currentNote;
        --semitones;
    } while (!(currentNote == generatedTransposition.second));
}
ASSERT_TRUE(elegia::Transposer::IsCorrectAnswer(semitones));
}

```

Опис: генерація транспозиції та її невгадування Коли: згенерований інтервал не збігається із запропонованою користувачем відповіддю Тоді: функція перевірки повертає false

```

TEST(ElegiaTransposerTest, WrongRange)
{
    // Повертає кілька нот.
    const auto generatedTransposition{
elegia::Transposer::GenerateTransposition(true) };

```

```

data::Note currentNote{ generatedTransposition.first };
std::int8_t semitones = 0;
// Підбір хибного значення.
// Якщо інтервалу немає – задаємо відповідь як 1. Інакше 0..
if (generatedTransposition.first == generatedTransposition.second)
{
    semitones = 1;
}
ASSERT_FALSE(elegia::Transposer::IsCorrectAnswer(semitones));
}

```

### 3.2. Тестування методом «чорної скриньки»

Оскільки Elegia має великий інтерфейс, для програми буде потрібний метод тестування, який базується на принципі тестування чорної скриньки. Тестування чорної скриньки використовується для перевірки функціональності програми на основі зовнішніх входних та вихідних даних без врахування внутрішньої реалізації. Тест кейси, продемонстровані на таблиці 3.1.

Таблиця 3.1 тест кейсів для Elegia

| Опис тестового випадку                     | Кроки для відтворення   | Очікуваний результат   | Статус  |
|--|---|--|---------|
| Відкриття режиму Identify Note             | Натиснути на бічне меню Identify Note   | Перед користувачем відкривається вікно старту IdetifyNote з однотонним відтінком                         | Пройшов |
| Вибір нот для початку режиму Identify Node | 1. Вибрати кількість нот від 5 до нескінченності.<br>2. Натиснути кнопку Begin. | Перед користувачем відкривається вікно тренажера, де потрібно вибрати ноту і натиснути на кнопку старту. | Пройшов |

## Продовження таблиці 3.1

|                                   |  |  |         |
|-----------------------------------|--|--|---------|
| Запуск вправи IdentifyNote        | Натиснути кнопку Begin   | Перед користувачем відкривається сторінка з вправами і він бачить всі елементи для тестування на ноти. | Пройшов |
| Регулювання звуку в режимі IN     | 1. Натисніть іконку звуку, щоб вимкнути звук. 2. Переміщуйте повзунок поруч із звуком.   | У користувача буде можливість змінювати звук на його розсуд  | Пройшов |
| Вихід з режиму IN                 | Натиснути кнопку exit, щоб вийти з режиму  | Вихід з режиму з перериванням ноти, що програється   | Пройшов |
| Вибір правильної відповіді для IN | 1. Послухати ноту для ідентифікації правильної відповіді. 2. Вибрати ноту, яка відповідає програній ноті. 3. Натиснути кнопку next для зарахування правильної відповіді. | При виборі правильної відповіді користувачеві додається до боарду правильних відповідей +1.            | Пройшов |
| Вибір неправильної відповіді IN   | 1. Послухати ноту для ідентифікації правильної відповіді. 2. Вибрати ноту, яка не відповідає програній ноті. 3. Натиснути кнопку next                                    | При виборі неправильної відповіді користувачеві додається помилка в live-board.                        | Пройшов |

## Продовження таблиці 3.1

|   |  |  |         |
|---|--|--|---------|
| Кінець вправи в IN                              | Відповісти на всі поставлені вправи для користувача. І натиснути на Try Again.                               | Користувачеві після закінчення вправи відкриється статистика правильних відповідей та не правильних з мотиваційним текстом. І натиснути кнопку спробувати знову. | Пройшов |
| Запуск вільного режиму TonalityBuilder          | 1. Відкрити у бічному меню TonalityBuilder. 2. Натисніть кнопку TOOL.  | 1. Відкрити у бічному меню TonalityBuilder. 2. Натисніть кнопку TOOL. Користувач після дій побачить інтерфейс з фортепіано та налаштуваннями для створення нот.  | Пройшов |
| Вибір правильної відповіді у TonalityBuilder    | 1. Прослухати згенеровану рандомну тональність. 2. Вибрати правильну відповідь на поставлену тональність.    | Користувачеві буде додаватися бал у LIVE-Борд правильних відповідей, якщо користувач відповідь правильно.  | Пройшов |
| Вибір не правильної відповіді у TonalityBuilder | 1. Прослухати згенеровану рандомну тональність. 2. Вибрати не правильну відповідь на поставлену тональність. | Користувачеві буде додаватися бал у LIVE-Борд не правильних відповідей, якщо користувач відповідь не правильно.  | Пройшов |

## Продовження таблиці 3.1

|   |   |  |         |
|---|---|--|---------|
| Вибір параметрів для створення тональності у вільному інструменті TonicityBuilder | 1. Вибрати тон для тональності. 2. Вибрати символ альтерації для тональності. 3. Вибрати ключ для тональності. 4. Натиснути Build для побудови тональності. | Користувач має перед собою вибір із трьох налаштувань, які він повинен вибрати для побудови тональності. І натиснути кнопку для побудови тональності.      | Пройшов |
| Вихід із режиму TonicityBuilder під час використання вільного інструменту.        | Натисніть на кнопку exit, щоб вийти з вільного інструменту.   | Користувач вийде з вільного інструменту тональності і цим прирве програну тональність.   | Пройшов |
| Програти тональності вільного режиму TonicityBuilder                              | 1. Натисніть кнопку Build. 2. Натиснути на play щоб програвся побудована користувачем тональність.  | Користувач після натискання на Play почує мелодію, що програється.   | Пройшов |
| Вибір двох налаштувань для створення мелодії TonicityBuilder                      | 1. Вибрати тонік. 2. Вибрати символ альтерації. 3. Натиснути Build, щоб побудувати тональність.   | Користувач вибравши двох налаштувань із трьох натискає на Build але нічого не відбувається так як бракує останнього налаштування для побудови тональності. | Пройшов |
| Запуск TonicityBuilder вправ  | 1. Вибери́ть кількість тональностей. 2. Натисніть кнопку на LEAR  | Користувач вибирає кількість тональностей від 5 до нескінченності та натискає на LEAR  | Пройшов |

## Продовження таблиці 3.1

|  |   |   |         |
|--|---|---|---------|
|  | для початку навчання.   | для старту вправ TonicityBuilder  |         |
| Тест TonicityBuilder (Звук)                  | Натиснути кнопку play для відігравання побудованою системою мелодії.  | Користувач повинен послухати мелодії або подивитися на фортопіано та побачити побудову точки для відгадування тональності.                            | Пройшов |
| Вправи TonicityBuilder                       | 1. Вибрати тон для тональності. 2. Вибрати символ альтерації для тональності. 3. Вибрати ключ для тональності. 4. Натиснути Build для побудови тональності. | Користувач має перед собою вибір із трьох налаштувань, які він повинен вибрати для побудови тональності. І натиснути кнопку для побудови тональності. | Пройшов |
| Вибір неправильної відповіді TonicityBuilder | 1. Вибрати тонік. 2. Вибрати символ альтерацій. 3. Вибрати ключ. 4. Натиснути на Next   | Користувач вводить неправильну відповідь на поставлену тональність та кількість спроб зменшується.  | Пройшов |
| Вибір правильної відповіді TonicityBuilder   | 1. Вибрати тонік. 2. Вибрати символ альтерацій. 3. Вибрати ключ. 4. Натиснути на Next   | Користувач вводить правильну відповідь на поставлену тональність і додається одиниця в live-board.  | Пройшов |

## Продовження таблиці 3.1

|   |  |   |         |
|---|--|---|---------|
| Управління гучністю звуку у режимі тренування TonicityBuilder | Управління повзунком біля play.  | Користувач має можливість зменшити гучність тональності, що програвється під себе.  | Пройшов |
| Закінчення вправи TonicityBuilder                             | Закінчити тест із обраною тональністю.   | Користувачеві відкривається сторінка результатів і кнопка, яка відправила його на початок режиму для вибору кількості тональностей. | Пройшов |
| Відкрити режим Transposer                                     | Вибрати у бічній панелі Transposer.  | Користувач перемикається в бічній панелі та бачить однотонний дизайн режиму   | Пройшов |
| Побудова транспозиції у Transpose                             | 1. Вибрати ступінь. 2. Вибрати символ альтерації. 3. Вибрати октаву. 4. Вибрати півтону. 5. Натиснути кнопку Transpose | Користувач вибирає налаштування для побудови транспозиції з його параметрами.   | Пройшов |
| Тест на програвання звуку інструменту транспозиції            | 1. Вибрати всі налаштування. 2. Зробити транспозицію. 3. Натиснути Play.   | Користувач може програти стільки разів транспозиції, доки йому потрібно буде індефікувати транспозицію.                             | Пройшов |
| Тест вправи Transposer  | Вибрати півтону  | Користувач вибирає півтони  | Пройшов |



Продовження таблиці 3.1

|  |  |   |         |
|--|--|---|---------|
|  |  | для зарахування правильного або неправильного тексту.   |         |
| Вибір правильної відповіді у Transposer    | Вибрати правильну відповідь для тестування.                  | Користувач вводить правильну відповідь завдяки програному звуку або точкам на фортепіон. При виборі правильної відповіді додає бал до live-board.       | Прийшов |
| Вибір не правильної відповіді у Transposer | Вибрати не правильну відповідь для тестування.               | Користувач вводить не правильну відповідь завдяки програному звуку або точкам на фортепіон. При виборі не правильної відповіді додає бал до live-board. | Прийшов |
| Завершення Transposer                      | Завершити всі задані транспозиції для отримання результатів. | Користувач закінчує всі задані транспозиції на початку тренування, і бачить результати правильних і неправильних відповідей із кнопкою почати спочатку. | Прийшов |

Висновки до пункту 3.

На етапі тестування чорним ящиком таск кейсів не було виявлено помилок з боку інтерфейсу для користувача. Зовнішня частина продукту досягає вимог, які потрібні користувачеві. Однотонні кольори та ненависні об'єкти.

## ВИСНОВОК

На даний час існує багато альтернативних програм, які можуть допомогти користувачеві вивчити транспозицію та навчитися визначати за звуком, який півтонь використовується в транспозиції. Використання такої програми дозволить користувачеві отримати впевненість і навички у роботі з транспозицією, а також мати доступ до інструментів, що дозволяють вправлятися в транспозиції та вибирати півтони. Під час розробки програми було приділено особливу увагу відображенню правильних півтонів на клавішах. Це має велике значення для музиканта, оскільки він повинен знати, на які клавіші натискати, щоб відтворити правильний звук. Також було приділено значну увагу звукам: неправильне відтворення звуків може спричинити неправильне засвоєння інформації музикантом та негативно позначитися на його майбутній кар'єрі.

Основним результатом проектування є інструмент, який точно визначатиме півтони та відтворюватиме звуки для користувача. Програмний продукт надасть можливість користувачу використовувати різні інструменти за своїм смаком у вільному режимі або під час тренування. Все, що потрібно користувачеві, - це вибрати, скільки разів він бажає пройти тренування з транспозицією нот і правильно відповідати на почуті звуки або самостійно розраховувати півтони за допомогою екрану.

У процесі роботи отримуємо транспозицію за допомогою фортепіано. Користувач може зробити свою транспозицію, вказавши знак альтерації, октаву, полутон та ступінь та продукт що може дати знання музикантові котрий буде користуватися програмою.

Завдяки такому підходу, кінцевий продукт повністю готовий для використання з користувачами з метою поліпшення сприйняття слуху та поглиблення бази знань з теми транспозиції в музиці.

## ЛІТЕРАТУРА

1. Інформація про QT фреймворк // (Електронний ресурс)  
<https://www.qt.io/blog/qt-6.2-lts-released>
2. І.В. Спосібін Елементарна теорія музики 1963 г.
3. Документація до QT // (Електронний ресурс)  
<https://doc.qt.io/qt.html>
4. Інформація про октаву // (Електронний ресурс)  
<http://electrovoice.com.ua/>
5. Інформація про Транспозицію // (Електронний ресурс)  
[https://uk.wikipedia.org/wiki/Транспозиція\\_\(музика\)](https://uk.wikipedia.org/wiki/Транспозиція_(музика))
6. Інформація про CMake // (Електронний ресурс)  
<https://uk.wikipedia.org/wiki/CMake>

## ДОДАТКИ

## ТЕКСТ ПРОГРАММИ

## mainwindow.ui:

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow">
<property name="windowModality">
<enum>Qt::NonModal</enum>
</property>
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>980</width>
<height>645</height>
</rect>
</property>
<property name="font">
<font>
<family>Arial Black</family>
<bold>true</bold>
</font>
</property>
<property name="tabletTracking">
<bool>false</bool>
</property>
<property name="windowTitle">
<string>Elegia</string>
</property>
<property name="windowIcon">
<iconset resource="../../resources/Images.qrc">
<normaloff>:/common/img_resources/iconwindow.png</normaloff>
f>:/common/img_resources/iconwindow.png</iconset>
</property>
<property name="autoFillBackground">
<bool>true</bool>
</property>
<widget class="QWidget" name="centralwidget">
<widget class="QLabel" name="label">
<property name="geometry">
<rect>
<x>-50</x>
<y>-10</y>
<width>1031</width>
<height>681</height>
</rect>
</property>
<property name="text">
<string>
</property>
<property name="pixmap">
<pixmap
resource="../../resources/Images.qrc">:/common/img_resources/MainBackground.jpg</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QGroupBox" name="NavigationGroupBox">
<property name="enabled">
<bool>true</bool>
</property>
<property name="geometry">
<rect>
<x>10</x>
<y>10</y>
<width>141</width>
<height>621</height>
</rect>
</property>
<property name="font">
<font>
<family>Arial Black</family>
<bold>true</bold>
</font>
</property>
<property name="styleSheet">
<string notr="true">QGroupBox {
border: 2px solid white;
}
</string>
</property>
<property name="alignment">
<set>Qt::AlignCenter</set>
</property>
<property name="flat">
<bool>false</bool>
</property>
<widget class="QPushButton" name="HomeButton">
<property name="geometry">
<rect>
<x>10</x>
<y>20</y>
<width>121</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton {
font: 10pt &quot;Comic Sans MS&quot;;
background-color: rgb(95, 95, 95);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(67, 67, 67);
border-right: 1px solid rgb(67, 67, 67);
border-bottom: 5px solid rgb(67, 67, 67);
border-radius: 8px;
}
QPushButton:hover {
font: 11pt &quot;Comic Sans MS&quot;;
background-color: rgb(145, 190, 100);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(110, 144, 76);
border-right: 1px solid rgb(110, 144, 76);
border-bottom: 5px solid rgb(110, 144, 76);
}
QPushButton:pressed {
font: 11pt &quot;Comic Sans MS&quot;;
background-color: rgb(120, 157, 83);
border-left: 1px solid rgb(110, 144, 76);
border-right: 1px solid rgb(110, 144, 76);
border-bottom: 5px solid rgb(110, 144, 76);
border-bottom: none;
}
</string>
</property>
<property name="text">
<string> Home</string>
</property>
<property name="icon">
<iconset
<normaloff>:/new/commonImages/img_resources/homePage_icon.png</normaloff>:/new/commonImages/img_resources/homePage_icon.png</iconset>
</property>
</widget>
<widget class="QPushButton" name="IdentifyNoteButton">
<property name="geometry">
<rect>
<x>10</x>
<y>90</y>
<width>121</width>
<height>51</height>
</rect>

```

```

    </property>
    <property name="styleSheet">
      <string notr="true">QPushButton{
        font: 9pt &quot;Comic Sans MS&quot;;
        background-color: rgb(95, 95, 95);
        border: none;
        color: rgb(226, 234, 216);
        border-left: 1px solid rgb(67, 67, 67);
        border-right: 1px solid rgb(67, 67, 67);
        border-bottom: 5px solid rgb(67, 67, 67);
        border-radius: 8px;
      }

QPushButton:hover{
  font: 10pt &quot;Comic Sans MS&quot;;
  background-color: rgb(145, 190, 100);
  border: none;
  color: rgb(226, 234, 216);
  border-left: 1px solid rgb(110, 144, 76);
  border-right: 1px solid rgb(110, 144, 76);
  border-bottom: 5px solid rgb(110, 144, 76);
}

QPushButton:pressed{
  font: 10pt &quot;Comic Sans MS&quot;;
  background-color: rgb(120, 157, 83);
  border-left: 1px solid rgb(110, 144, 76);
  border-right: 1px solid rgb(110, 144, 76);
  border-bottom: 5px solid rgb(110, 144, 76);
  border-bottom: none;
}
</string>
  </property>
  <property name="text">
    <string> Identify Note</string>
  </property>
  <property name="icon">
    <iconset>

<normaloff>:/new/commonImages/img_resources/musicPage_icon
.png</normaloff>:/new/commonImages/img_resources/musicPage
_icon.png</iconset>
  </property>
</widget>
<widget class="QPushButton" name="TonalityBuilderButton">
  <property name="geometry">
    <rect>
      <x>10</x>
      <y>160</y>
      <width>121</width>
      <height>51</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton{
      font: 9pt &quot;Comic Sans MS&quot;;
      background-color: rgb(95, 95, 95);
      border: none;
      color: rgb(226, 234, 216);
      border-left: 1px solid rgb(67, 67, 67);
      border-right: 1px solid rgb(67, 67, 67);
      border-bottom: 5px solid rgb(67, 67, 67);
      border-radius: 8px;
    }

QPushButton:hover{
  font: 10pt &quot;Comic Sans MS&quot;;
  background-color: rgb(145, 190, 100);
  border: none;
  color: rgb(226, 234, 216);
  border-left: 1px solid rgb(110, 144, 76);
  border-right: 1px solid rgb(110, 144, 76);
  border-bottom: 5px solid rgb(110, 144, 76);
}

QPushButton:pressed{
  font: 10pt &quot;Comic Sans MS&quot;;
  background-color: rgb(120, 157, 83);

```

```

    border-left: 1px solid rgb(110, 144, 76);
    border-right: 1px solid rgb(110, 144, 76);
    border-bottom: 5px solid rgb(110, 144, 76);
    border-bottom: none;
  }
</string>
  </property>
  <property name="text">
    <string>Tonality Builder</string>
  </property>
</widget>
<widget class="QPushButton" name="TransposerButton">
  <property name="geometry">
    <rect>
      <x>10</x>
      <y>230</y>
      <width>121</width>
      <height>51</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton{
      font: 9pt &quot;Comic Sans MS&quot;;
      background-color: rgb(95, 95, 95);
      border: none;
      color: rgb(226, 234, 216);
      border-left: 1px solid rgb(67, 67, 67);
      border-right: 1px solid rgb(67, 67, 67);
      border-bottom: 5px solid rgb(67, 67, 67);
      border-radius: 8px;
    }

QPushButton:hover{
  font: 10pt &quot;Comic Sans MS&quot;;
  background-color: rgb(145, 190, 100);
  border: none;
  color: rgb(226, 234, 216);
  border-left: 1px solid rgb(110, 144, 76);
  border-right: 1px solid rgb(110, 144, 76);
  border-bottom: 5px solid rgb(110, 144, 76);
}

QPushButton:pressed{
  font: 10pt &quot;Comic Sans MS&quot;;
  background-color: rgb(120, 157, 83);
  border-left: 1px solid rgb(110, 144, 76);
  border-right: 1px solid rgb(110, 144, 76);
  border-bottom: 5px solid rgb(110, 144, 76);
  border-bottom: none;
}
</string>
  </property>
  <property name="text">
    <string>Transposer</string>
  </property>
</widget>
<widget class="QPushButton" name="aboutUsButton">
  <property name="geometry">
    <rect>
      <x>26</x>
      <y>570</y>
      <width>91</width>
      <height>41</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton{
      font: 9pt &quot;Comic Sans MS&quot;;
      background-color: rgb(95, 95, 95);
      border: none;
      color: rgb(226, 234, 216);
      border-left: 1px solid rgb(67, 67, 67);
      border-right: 1px solid rgb(67, 67, 67);
      border-bottom: 5px solid rgb(67, 67, 67);
      border-radius: 8px;
    }

QPushButton:hover{

```

```

font: 10pt "Comic Sans MS";
background-color: rgb(145, 190, 100);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(110, 144, 76);
border-right: 1px solid rgb(110, 144, 76);
border-bottom: 5px solid rgb(110, 144, 76);
}

QPushButton:pressed {
font: 10pt "Comic Sans MS";
background-color: rgb(120, 157, 83);
border-left: 1px solid rgb(110, 144, 76);
border-right: 1px solid rgb(110, 144, 76);
border-bottom: 5px solid rgb(110, 144, 76);
border-bottom: none;
}
</string>
<property>
<property name="text">
<string>About us</string>
</property>
</widget>
</widget>
<widget class="QGroupBox" name="MainPanelGroupBox">
<property name="enabled">
<bool>true</bool>
</property>
<property name="geometry">
<rect>
<x>160</x>
<y>8</y>
<width>821</width>
<height>621</height>
</rect>
</property>
<property name="font">
<font>
<family>Arial Black</family>
<bold>true</bold>
</font>
</property>
<property name="mouseTracking">
<bool>false</bool>
</property>
<property name="autoFillBackground">
<bool>false</bool>
</property>
<property name="styleSheet">
<string notr="true">border-radius: 0px;</string>
</property>
<property name="inputMethodHints">
<set>Qt::ImhNone</set>
</property>
<property name="alignment">
<set>Qt::AlignCenter</set>
</property>
<property name="flat">
<bool>false</bool>
</property>
<property name="checkable">
<bool>false</bool>
</property>
<widget class="QStackedWidget" name="stackedWidget">
<property name="geometry">
<rect>
<x>0</x>
<y>10</y>
<width>811</width>
<height>631</height>
</rect>
</property>
<property name="autoFillBackground">
<bool>false</bool>
</property>
<property name="currentIndex">
<number>4</number>
</property>
</widget class="QWidget" name="homePage">
<widget class="QLabel" name="label_3">
<property name="geometry">
<rect>
<x>480</x>
<y>230</y>
<width>161</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font: 900 22pt "Arial Black";
color: rgb(255, 255, 255);</string>
</property>
<property name="text">
<string>Welcome!</string>
</property>
</widget>
<widget class="QLabel" name="label_10">
<property name="geometry">
<rect>
<x>540</x>
<y>350</y>
<width>31</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font: 15pt "Arial";
</string>
</property>
<property name="text">
<string>△</string>
</property>
</widget>
<widget class="QLabel" name="label_4">
<property name="geometry">
<rect>
<x>410</x>
<y>380</y>
<width>291</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font: 12pt "Yu Gothic UI";
color: rgb(255, 255, 255);</string>
</property>
<property name="text">
<string>&lt;html&gt;&lt;head&gt;&lt;body&gt;&lt;p
align="center"&gt;&gt;Choose the category from
the&lt;br&gt;left bar to begin
with!&lt;p&gt;&lt;/body&gt;&lt;/html&gt;</string>
</property>
</widget>
<widget class="QLabel" name="label_11">
<property name="geometry">
<rect>
<x>390</x>
<y>270</y>
<width>341</width>
<height>61</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font: 12pt "Yu Gothic UI";
color: rgb(255, 255, 255);</string>
</property>
<property name="text">
<string>&lt;html&gt;&lt;head&gt;&lt;body&gt;&lt;p
align="center"&gt;&gt;This is
&amp;"ELEGIA&quot; - the exciting tool that
helps&lt;br&gt;many people to become educated
musician&lt;br&gt;with music theory and
practice!&lt;p&gt;&lt;/body&gt;&lt;/html&gt;</string>
</property>
</widget>
<widget class="QLabel" name="HomePageMainImage">

```



```

<property name="geometry">
<rect>
<x>220</x>
<y>-10</y>
<width>371</width>
<height>201</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="./resources/Images.qrc">:/common/img_resources/hom
ePage_MainImageTop.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
<property name="alignment">
<set>Qt::AlignCenter</set>
</property>
<property name="wordWrap">
<bool>>false</bool>
</property>
</widget>
<widget class="QLabel" name="label_2">
<property name="geometry">
<rect>
<x>10</x>
<y>150</y>
<width>341</width>
<height>461</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="./resources/Images.qrc">:/common/img_resources/mai
nwoman.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
</widget>
<widget class="QWidget" name="notesIdentifierStartPage">
<widget class="QLabel" name="NotesLabelPng">
<property name="geometry">
<rect>
<x>-10</x>
<y>50</y>
<width>531</width>
<height>451</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true"/>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="./resources/Images.qrc">:/noteIdentifier/img_resources/
notesPage2.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="label_5">
<property name="geometry">
<rect>
<x>400</x>
<y>130</y>
<width>321</width>
<height>191</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font: 12pt &quot;Yu Gothic UI&quot;;
color: rgb(255, 255, 255);</string>
</property>
<property name="text">
<string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p
align=&quot;center&quot;&gt;This training tool will help
you&lt;/p&gt;&lt;p align=&quot;center&quot;&gt;to remember
the notes on the&lt;/p&gt;&lt;p
align=&quot;center&quot;&gt;note sheet.&lt;/p&gt;&lt;p
align=&quot;center&quot;&gt;Select amount of the
notes&lt;/p&gt;&lt;p align=&quot;center&quot;&gt;you wish to
identify, and press&lt;/p&gt;&lt;p
align=&quot;center&quot;&gt;BEGIN button
below.&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
</property>
</widget>
<widget class="QPushButton" name="beginButtonNI">
<property name="enabled">
<bool>true</bool>
</property>
<property name="geometry">
<rect>
<x>450</x>
<y>380</y>
<width>221</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton {
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 85, 0);
border: 1px solid black;
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:hover {
background-color: rgb(8, 126, 0);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:pressed {
background-color: rgb(8, 126, 0);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
border-bottom: none;
}
</string>
</property>
<property name="text">
<string>Begin</string>
</property>
<property name="checkable">
<bool>>false</bool>
</property>
<property name="flat">
<bool>>false</bool>
</property>
</widget>
<widget class="QLabel" name="label_7">
<property name="geometry">
<rect>

```

```

        <x>460</x>
        <y>80</y>
        <width>211</width>
        <height>41</height>
    </rect>
</property>
<property name="styleSheet">
    <string notr="true">font: 900 22pt &quot;Arial Black&quot;;
color: rgb(255, 255, 255);</string>
</property>
<property name="text">
    <string>Identify Note</string>
</property>
</widget>
<widget class="QComboBox"
name="maxNotesSelectorComboBox">
    <property name="geometry">
        <rect>
            <x>450</x>
            <y>460</y>
            <width>220</width>
            <height>51</height>
        </rect>
    </property>
    <property name="styleSheet">
        <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
    background-color: rgb(144, 144, 144);
    border-radius: 10px;
    border-color: beige;
    font: bold 14px;
    min-width: 10em;
    padding: 10px;
}

QComboBox:pressed {
    border-style: inset;
}

QComboBox::item:hover {
    background-color: rgb(170, 170, 255);
    color: rgb(0, 0, 0);
}</string>
    </property>
    <property name="currentText">
        <string/>
    </property>
    <property name="placeholderText">
        <string>Choose notes amount...</string>
    </property>
    <item>
        <property name="text">
            <string>5 Notes</string>
        </property>
    </item>
    <item>
        <property name="text">
            <string>10 Notes</string>
        </property>
    </item>
    <item>
        <property name="text">
            <string>20 Notes</string>
        </property>
    </item>
    <property name="text">

```

```

        <string>Endless</string>
    </property>
</item>
</widget>
</widget>
<widget class="QWidget" name="notesIdentifierMainPage">
    <widget class="QPushButton" name="exitButton">
        <property name="geometry">
            <rect>
                <x>20</x>
                <y>540</y>
                <width>191</width>
                <height>51</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">QPushButton {
background-color: rgb(202, 26, 26);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:hover {
    background-color: rgb(247, 0, 0);
    font-family: 'Montserrat';
    font-size: 21px;
    font-weight: bold;
    color: #ffffff;
    padding: 9px 25px;
    border-radius: 20px;
    border: 1px solid black;
    border-left: 1px solid rgb(170, 0, 0);
    border-right: 1px solid rgb(170, 0, 0);
    border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:pressed {
    background-color: rgb(247, 0, 0);
    font-family: 'Montserrat';
    font-size: 21px;
    font-weight: bold;
    color: #ffffff;
    padding: 9px 25px;
    border-radius: 20px;
    border: 1px solid black;
    border-left: 1px solid rgb(170, 0, 0);
    border-right: 1px solid rgb(170, 0, 0);
    border-bottom: 5px solid rgb(170, 0, 0);
}</string>
        </property>
        <property name="text">
            <string>✕ EXIT</string>
        </property>
        <property name="iconSize">
            <size>
                <width>10</width>
                <height>10</height>
            </size>
        </property>
    </widget>
    <widget class="QPushButton" name="nextButton">
        <property name="geometry">
            <rect>
                <x>600</x>
                <y>540</y>
                <width>191</width>
                <height>51</height>
            </rect>
        </property>

```

```

        <property name="styleSheet">
        <string notr="true">QPushButton {
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 85, 0);
border: 1px solid black;
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:hover {
background-color: rgb(8, 126, 0);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:pressed {
background-color: rgb(8, 126, 0);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
border-bottom: none;
} </string>
        </property>
        <property name="text">
        <string>Next ►</string>
        </property>
        </widget>
        <widget class="QComboBox"
name="noteSelectorComboBox">
        <property name="geometry">
        <rect>
        <x>320</x>
        <y>510</y>
        <width>191</width>
        <height>61</height>
        </rect>
        </property>
        <property name="styleSheet">
        <string notr="true">font-family: 'Montserrat';
font-size: 18px;
font-weight: bold;
color: #ffffff;
padding: 15px 25px;
border-radius: 6px;
border: 1px solid black;
background-color: rgb(0, 0, 255);</string>
        </property>
        <property name="currentText">
        <string/>
        </property>
        <property name="placeholderText">
        <string>Select note...</string>
        </property>
        <item>
        <property name="text">
        <string>C (do)</string>
        </property>
        </item>
        <item>
        <property name="text">
        <string>D (re)</string>
        </property>
        </item>
        <item>
        <property name="text">
        <string>E (mi)</string>
        </property>
        </item>
        </item>
        <property name="text">
        <string>F (fa)</string>
        </property>
        </item>
        <item>
        <property name="text">
        <string>G (sol)</string>
        </property>
        </item>
        <item>
        <property name="text">
        <string>A (la)</string>
        </property>
        </item>
        <item>
        <property name="text">
        <string>B (si)</string>
        </property>
        </item>
        </widget>
        <widget class="QLabel" name="totalAnswersLabel">
        <property name="geometry">
        <rect>
        <x>30</x>
        <y>30</y>
        <width>111</width>
        <height>41</height>
        </rect>
        </property>
        <property name="styleSheet">
        <string notr="true">font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: white;
padding: 11px 25px;
border-radius: 15px;
border: 1px solid black;
background-color: rgb(124, 124, 124);</string>
        </property>
        <property name="text">
        <string>0/20</string>
        </property>
        </widget>
        <widget class="QLabel" name="correctAnswersLabel">
        <property name="geometry">
        <rect>
        <x>690</x>
        <y>30</y>
        <width>111</width>
        <height>41</height>
        </rect>
        </property>
        <property name="styleSheet">
        <string notr="true">font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: rgb(0, 206, 0);
padding: 11px 25px;
border-radius: 15px;
border: 1px solid black;
background-color: rgb(124, 124, 124);</string>
        </property>
        <property name="text">
        <string>0/20</string>
        </property>
        </widget>
        <widget class="QGroupBox" name="groupBox">
        <property name="geometry">
        <rect>
        <x>20</x>
        <y>80</y>
        <width>781</width>
        <height>411</height>
        </rect>
        </property>
        <property name="styleSheet">
        <string notr="true">background-color: rgb(255, 255, 255);
border-radius: 15px;</string>

```

```

</property>
<property name="title">
<string/>
</property>
<widget class="QLabel" name="NotesSheet">
<property name="geometry">
<rect>
<x>70</x>
<y>70</y>
<width>641</width>
<height>291</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="../../resources/Images.qrc">:/noteIdentifier/img_resources/
stave.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="NoteOnSheet">
<property name="geometry">
<rect>
<x>350</x>
<y>120</y>
<width>140</width>
<height>140</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">background: transparent;</string>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="../../resources/Images.qrc">:/noteIdentifier/img_resources/
note.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
</widget>
<widget class="QPushButton" name="soundControl">
<property name="geometry">
<rect>
<x>260</x>
<y>20</y>
<width>51</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton{
background-color: rgb(130, 170, 90);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(110, 144, 76);
border-right: 1px solid rgb(110, 144, 76);
border-bottom: 5px solid rgb(110, 144, 76);
border-radius: 8px;
}

QPushButton:pressed{
background-color: rgb(120, 157, 83);
border-left: 1px solid rgb(110, 144, 76);
border-right: 1px solid rgb(110, 144, 76);
border-bottom: 5px solid rgb(110, 144, 76);
border-bottom: none;
}
</string>
</property>
<property name="text">
<string/>
</property>
<property name="icon">
<iconset resource="../../resources/Images.qrc">

<normaloff>:/common/img_resources/volume.png</normaloff>:/c
ommon/img_resources/volume.png</iconset>
</property>
<property name="iconSize">
<size>
<width>32</width>
<height>32</height>
</size>
</property>
</widget>
<widget class="QSlider" name="audioVolumeSlide">
<property name="geometry">
<rect>
<x>320</x>
<y>30</y>
<width>251</width>
<height>31</height>
</rect>
</property>
<property name="maximum">
<number>100</number>
</property>
<property name="sliderPosition">
<number>100</number>
</property>
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
</widget>
</widget>
<widget class="QWidget" name="notesIdentifierResultPage">
<widget class="QLabel" name="results">
<property name="geometry">
<rect>
<x>350</x>
<y>150</y>
<width>111</width>
<height>91</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 900 20pt &quot;Arial Black&quot;;
}
</string>
</property>
<property name="text">
<string>Results</string>
</property>
</widget>
<widget class="QPushButton" name="resultPageExitButton">
<property name="geometry">
<rect>
<x>600</x>
<y>540</y>
<width>181</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton{
background-color: rgb(202, 26, 26);
font-family: 'Montserrat';

```

```

font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:hover{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:pressed{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

</string>
</property>
<property name="text">
<string>TRY AGAIN</string>
</property>
</widget>
<widget class="QTextEdit" name="resultText">
<property name="geometry">
<rect>
<x>200</x>
<y>330</y>
<width>411</width>
<height>61</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QTextEdit{
color: rgb(255, 255, 255);
}
</string>
</property>
<property name="readOnly">
<bool>true</bool>
</property>
<property name="html">
<string>&lt;!DOCTYPE HTML PUBLIC &quot;-
//W3C//DTD HTML 4.0//EN&quot;
&quot;http://www.w3.org/TR/REC-html40/strict.dtd&quot;&gt;
&lt;html&gt;&lt;head&gt;&lt;meta name=&quot;richtext&quot;
content=&quot;l&quot; /&gt;&lt;meta charset=&quot;utf-8&quot;
/&gt;&lt;style type=&quot;text/css&quot;&gt;
p, li { white-space: pre-wrap; }
hr { height: 1px; border-width: 0; }
li.unchecked::marker { content: &quot;\2610&quot;; }
li.checked::marker { content: &quot;\2612&quot;; }
&lt;/style&gt;&lt;/head&gt;&lt;body style=&quot;font-
family:'Segoe UI'; font-size:9pt; font-weight:400; font-
style:normal&quot;&gt;
&lt;p align=&quot;center&quot; style=&quot;margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot;font-

```

```

size:14pt;&quot;&gt;Your sheet music skills will improve with
each attempt, keep up the good
work.&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string
>
</property>
</widget>
<widget class="QLabel" name="totalAnswersResult">
<property name="geometry">
<rect>
<x>320</x>
<y>280</y>
<width>71</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 14pt &quot;MV Boli&quot;;
}
</string>
</property>
<property name="text">
<string>Total:</string>
</property>
</widget>
<widget class="QLabel" name="correctAnswersResult">
<property name="geometry">
<rect>
<x>320</x>
<y>230</y>
<width>81</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(0, 255, 0);
font: 14pt &quot;MV Boli&quot;;
}
</string>
</property>
<property name="text">
<string>Correct: </string>
</property>
</widget>
<widget class="QLabel" name="totalAnswersResultValue">
<property name="geometry">
<rect>
<x>420</x>
<y>280</y>
<width>71</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 14pt &quot;MV Boli&quot;;
}
</string>
</property>
<property name="text">
<string>123</string>
</property>
</widget>
<widget class="QLabel" name="correctAnswersResultValue">
<property name="geometry">
<rect>
<x>420</x>
<y>230</y>
<width>71</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(0, 255, 0);
font: 14pt &quot;MV Boli&quot;;
}
</string>
</property>

```

```

        <property name="text">
        <string>123</string>
    </property>
</widget>
</widget>
<widget class="QWidget" name="tonalityBuilderStartPage">
    <widget class="QTextEdit" name="mainTextTB">
        <property name="geometry">
            <rect>
                <x>230</x>
                <y>260</y>
                <width>361</width>
                <height>121</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">QTextEdit {
                color: rgb(255, 255, 255);
            }</string>
        </property>
        <property name="readOnly">
            <bool>true</bool>
        </property>
        <property name="html">
            <string>&lt;!DOCTYPE HTML PUBLIC &quot;-
//W3C//DTD HTML 4.0//EN&quot;
&quot;http://www.w3.org/TR/REC-html40/strict.dtd&quot;&gt;
&lt;html&gt;&lt;head&gt;&lt;meta name=&quot;richtext&quot;
content=&quot;1&quot; /&gt;&lt;meta charset=&quot;utf-8&quot;
/ &gt;&lt;style type=&quot;text/css&quot;&gt;
p, li { white-space: pre-wrap; }
hr { height: 1px; border-width: 0; }
li.unchecked::marker { content: &quot;\2610&quot;; }
li.checked::marker { content: &quot;\2612&quot;; }
&lt;/style&gt;&lt;/head&gt;&lt;body style=&quot;font-
family:'Segoe UI'; font-size:9pt; font-weight:400; font-
style:normal&quot;&gt;
&lt;p align=&quot;center&quot; style=&quot;margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot;font-
size:14pt;&quot;&gt;You can visualize the chosen tonalities on
piano keys.&lt;/span&gt;&lt;/p&gt;
&lt;p align=&quot;center&quot; style=&quot;margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot;font-
size:14pt;&quot;&gt;You can choose between learning mode and
tool mode
below.&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
        </property>
    </widget>
    <widget class="QLabel" name="tonalityBuilderHeader">
        <property name="geometry">
            <rect>
                <x>290</x>
                <y>220</y>
                <width>301</width>
                <height>41</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">font: 900 22pt &quot;Arial Black&quot;;
color: rgb(255, 255, 255);</string>
        </property>
        <property name="text">
            <string>Tonality Builder </string>
        </property>
    </widget>
    <widget class="QPushButton" name="toolButtonTB">
        <property name="geometry">
            <rect>
                <x>570</x>
                <y>500</y>
                <width>191</width>
                <height>61</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">QPushButton {
                font-family: 'Montserrat';
                font-size: 21px;
                font-weight: bold;
                color: #ffffff;
                padding: 11px 25px;
                border-radius: 20px;
                background-color: rgb(0, 85, 0);
                border: 1px solid black;
                border-left: 1px solid rgb(26, 65, 0);
                border-right: 1px solid rgb(26, 65, 0);
                border-bottom: 5px solid rgb(26, 65, 0);
            }

QPushButton:hover {
                background-color: rgb(8, 126, 0);
                border: none;
                color: rgb(226, 234, 216);
                border-left: 1px solid rgb(26, 65, 0);
                border-right: 1px solid rgb(26, 65, 0);
                border-bottom: 5px solid rgb(26, 65, 0);
            }

QPushButton:pressed {
                background-color: rgb(8, 126, 0);
                border-left: 1px solid rgb(26, 65, 0);
                border-right: 1px solid rgb(26, 65, 0);
                border-bottom: 5px solid rgb(26, 65, 0);
                border-bottom: none;
            }</string>
        </property>
        <property name="text">
            <string>TOOL</string>
        </property>
    </widget>
    <widget class="QPushButton" name="learnButtonTB">
        <property name="geometry">
            <rect>
                <x>60</x>
                <y>500</y>
                <width>191</width>
                <height>61</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">QPushButton {
                font-family: 'Montserrat';
                font-size: 21px;
                font-weight: bold;
                color: #ffffff;
                padding: 11px 25px;
                border-radius: 20px;
                background-color: rgb(0, 144, 255);
                border: 1px solid black;
                border-left: 1px solid rgb(0, 98, 255);
                border-right: 1px solid rgb(0, 98, 255);
                border-bottom: 5px solid rgb(0, 98, 255);
            }

QPushButton:hover {
                background-color: rgb(0, 144, 255);
                border: none;
                color: rgb(226, 234, 216);
                border-left: 1px solid rgb(0, 98, 255);
                border-right: 1px solid rgb(0, 98, 255);
                border-bottom: 5px solid rgb(0, 98, 255);
            }

QPushButton:pressed {
                background-color: rgb(0, 144, 255);
                border-left: 1px solid rgb(0, 98, 255);
                border-right: 1px solid rgb(0, 98, 255);
                border-bottom: 5px solid rgb(0, 98, 255);
                border-bottom: none;
            }</string>
        </property>
        <property name="text">
            <string>LEARN</string>
        </property>
    </widget>

```

```

        </property>
    </widget>
    <widget class="QLabel" name="imageTB">
        <property name="geometry">
            <rect>
                <x>126</x>
                <y>-9</y>
                <width>571</width>
                <height>301</height>
            </rect>
        </property>
        <property name="text">
            <string>
        </property>
        <property name="pixmap">
            <pixmap
resource="../../resources/Images.qrc">:/common/img_resources/pian
o.png</pixmap>
        </property>
        <property name="scaledContents">
            <bool>true</bool>
        </property>
    </widget>
    <widget class="QComboBox"
name="tonalitiesSelectorComboBox">
        <property name="geometry">
            <rect>
                <x>40</x>
                <y>450</y>
                <width>231</width>
                <height>41</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover{
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover{
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
}
</string>
        </property>
        <property name="currentIndex">
            <number>-1</number>
        </property>
        <property name="placeholderText">
            <string>Choose tonalities amount...</string>
        </property>
        <item>
            <property name="text">
                <string>5 Tonalities</string>
            </property>
        </item>
        <item>
            <property name="text">
                <string>10 Tonalities</string>
            </property>
        </item>
    </widget>
    <widget class="QWidget" name="tonalityBuilderMainPage">
        <widget class="QComboBox" name="tonicSelectorTB">
            <property name="geometry">
                <rect>
                    <x>290</x>
                    <y>450</y>
                    <width>241</width>
                    <height>41</height>
                </rect>
            </property>
            <property name="styleSheet">
                <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover{
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover{
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
}
</string>
            </property>
            <property name="currentIndex">
                <number>-1</number>
            </property>
            <property name="placeholderText">
                <string>Choose a tonic...</string>
            </property>
            <item>
                <property name="text">
                    <string>C</string>
                </property>
            </item>
            <item>
                <property name="text">
                    <string>D</string>
                </property>
            </item>
            <item>
                <property name="text">
                    <string>E</string>
                </property>
            </item>
            <item>
                <property name="text">
                    <string>F</string>
                </property>
            </item>
        </widget>
    </widget>

```

```

    </property>
  </item>
</item>
  <property name="text">
    <string>G</string>
  </property>
</item>
<item>
  <property name="text">
    <string>A</string>
  </property>
</item>
<item>
  <property name="text">
    <string>B</string>
  </property>
</item>
</widget>
<widget class="QComboBox" name="modeSelectorTB">
  <property name="geometry">
    <rect>
      <x>290</x>
      <y>550</y>
      <width>241</width>
      <height>41</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover{
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover{
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
}</string>
  </property>
  <property name="placeholderText">
    <string>Choose alteration sign...</string>
  </property>
  <item>
    <property name="text">
      <string>None</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>#</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>b</string>
    </property>
  </item>
</widget>
<widget class="QPushButton" name="playTB">
  <property name="geometry">
    <rect>
      <x>200</x>
      <y>380</y>
      <width>151</width>
      <height>51</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton{
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 144, 255);
border: 1px solid black;
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);

```



```

}

QPushButton:hover{
    background-color: rgb(0, 144, 255);
    border: none;
    color: rgb(226, 234, 216);
    border-left: 1px solid rgb(0, 98, 255);
    border-right: 1px solid rgb(0, 98, 255);
    border-bottom: 5px solid rgb(0, 98, 255);
}

QPushButton:pressed{
    background-color: rgb(0, 144, 255);
    border-left: 1px solid rgb(0, 98, 255);
    border-right: 1px solid rgb(0, 98, 255);
    border-bottom: 5px solid rgb(0, 98, 255);
    border-bottom: none;
} </string>
</property>
<property name="text">
<string>Play ▶</string>
</property>
</widget>
<widget class="QPushButton" name="buildTB">
<property name="geometry">
<rect>
<x>600</x>
<y>550</y>
<width>191</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton{
    font-family: 'Montserrat';
    font-size: 21px;
    font-weight: bold;
    color: #ffffff;
    padding: 11px 25px;
    border-radius: 20px;
    background-color: rgb(0, 85, 0);
    border: 1px solid black;
    border-left: 1px solid rgb(26, 65, 0);
    border-right: 1px solid rgb(26, 65, 0);
    border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:hover{
    background-color: rgb(8, 126, 0);
    border: none;
    color: rgb(226, 234, 216);
    border-left: 1px solid rgb(26, 65, 0);
    border-right: 1px solid rgb(26, 65, 0);
    border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:pressed{
    background-color: rgb(8, 126, 0);
    border-left: 1px solid rgb(26, 65, 0);
    border-right: 1px solid rgb(26, 65, 0);
    border-bottom: 5px solid rgb(26, 65, 0);
    border-bottom: none;
} </string>
</property>
<property name="text">
<string>Build</string>
</property>
</widget>
<widget class="QLabel" name="pianoImage">
<property name="geometry">
<rect>
<x>30</x>
<y>80</y>
<width>751</width>
<height>281</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
    border-radius: 5px;
    border: 2px solid rgb(57, 57, 57);
} </string>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="..resources/Images.qrc">:/common/img_resources/pianoActivate.jif</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="C3">
<property name="geometry">
<rect>
<x>33</x>
<y>321</y>
<width>31</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="..resources/Images.qrc">:/common/img_resources/dot1.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="D3">
<property name="geometry">
<rect>
<x>70</x>
<y>321</y>
<width>31</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="..resources/Images.qrc">:/common/img_resources/dot1.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="E3">
<property name="geometry">
<rect>
<x>105</x>
<y>321</y>
<width>31</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource="..resources/Images.qrc">:/common/img_resources/dot1.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>

```









```

        <ixmap
resource="../../resources/Images.qrc">:/common/img_resources/UPd
ot1.png</ixmap>
    </property>
    <property name="scaledContents">
    <bool>true</bool>
    </property>
</widget>
<widget class="QLabel" name="Fs5">
<property name="geometry">
<rect>
    <x>657</x>
    <y>210</y>
    <width>21</width>
    <height>21</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="ixmap">
<ixmap
resource="../../resources/Images.qrc">:/common/img_resources/UPd
ot1.png</ixmap>
    </property>
    <property name="scaledContents">
    <bool>true</bool>
    </property>
</widget>
<widget class="QLabel" name="Gs5">
<property name="geometry">
<rect>
    <x>698</x>
    <y>210</y>
    <width>21</width>
    <height>21</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="ixmap">
<ixmap
resource="../../resources/Images.qrc">:/common/img_resources/UPd
ot1.png</ixmap>
    </property>
    <property name="scaledContents">
    <bool>true</bool>
    </property>
</widget>
<widget class="QLabel" name="As5">
<property name="geometry">
<rect>
    <x>740</x>
    <y>210</y>
    <width>21</width>
    <height>21</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="ixmap">
<ixmap
resource="../../resources/Images.qrc">:/common/img_resources/UPd
ot1.png</ixmap>
    </property>
    <property name="scaledContents">
    <bool>true</bool>
    </property>
</widget>
<widget class="QLabel" name="totalAnswersLabel_2">
<property name="geometry">
<rect>
    <x>30</x>
    <y>20</y>
    <width>111</width>
    <height>41</height>

```

```

    </rect>
</property>
<property name="styleSheet">
<string notr="true">font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: white;
padding: 11px 25px;
border-radius: 15px;
border: 1px solid black;
background-color: rgb(124, 124, 124);</string>
</property>
<property name="text">
<string>0/20</string>
</property>
</widget>
<widget class="QLabel" name="correctAnswersLabel_2">
<property name="geometry">
<rect>
    <x>670</x>
    <y>20</y>
    <width>111</width>
    <height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: rgb(0, 206, 0);
padding: 11px 25px;
border-radius: 15px;
border: 1px solid black;
background-color: rgb(124, 124, 124);</string>
</property>
<property name="text">
<string>0/20</string>
</property>
</widget>
<widget class="QPushButton" name="exitButton_2">
<property name="geometry">
<rect>
    <x>30</x>
    <y>550</y>
    <width>191</width>
    <height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton{
background-color: rgb(202, 26, 26);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton: hover{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

```

```

QPushButton:pressed {
    background-color: rgb(247, 0, 0);
    font-family: 'Montserrat';
    font-size: 21px;
    font-weight: bold;
    color: #ffffff;
    padding: 9px 25px;
    border-radius: 20px;
    border: 1px solid black;
    border-left: 1px solid rgb(170, 0, 0);
    border-right: 1px solid rgb(170, 0, 0);
    border-bottom: 5px solid rgb(170, 0, 0);
} </string>
</property>
<property name="text">
<string> ✕ EXIT</string>
</property>
<property name="iconSize">
<size>
<width>10</width>
<height>10</height>
</size>
</property>
</widget>
<widget class="QSlider" name="audioVolumeSlide_2">
<property name="geometry">
<rect>
<x>370</x>
<y>390</y>
<width>251</width>
<height>31</height>
</rect>
</property>
<property name="maximum">
<number>100</number>
</property>
<property name="sliderPosition">
<number>100</number>
</property>
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
</widget>
</widget>
<widget class="QWidget" name="tonalityBuilderResultPage">
<widget class="QTextEdit" name="resultText_2">
<property name="geometry">
<rect>
<x>200</x>
<y>330</y>
<width>411</width>
<height>61</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QTextEdit {
    color: rgb(255, 255, 255);
} </string>
</property>
<property name="readOnly">
<bool>true</bool>
</property>
<property name="html">
<string>&lt;!DOCTYPE HTML PUBLIC &quot;-
//W3C//DTD HTML 4.0//EN&quot;
&quot;http://www.w3.org/TR/REC-html40/strict.dtd&quot;&gt;
&lt;html&gt;&lt;head&gt;&lt;meta name=&quot;richtext&quot;
content=&quot;l&quot; /&gt;&lt;meta charset=&quot;utf-8&quot;
/&gt;&lt;style type=&quot;text/css&quot;&gt;
p, li { white-space: pre-wrap; }
hr { height: 1px; border-width: 0; }
li.unchecked::marker { content: &quot;2610&quot;; }
li.checked::marker { content: &quot;2612&quot;; }
&lt;/style&gt;&lt;/head&gt;&lt;body style=&quot; font-
family:'Segoe UI'; font-size:9pt; font-weight:400; font-
style:normal&quot;&gt;
&lt;p align=&quot;center&quot; style=&quot; margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-

```

```

indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot; font-
size:14pt;&quot;&gt;Your tonality knowledge will improve with
each attempt, keep up the good
work.&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;&lt;/string
>

```

```

</property>
</widget>
<widget class="QLabel"
name="correctAnswersResultValue_2">
<property name="geometry">
<rect>
<x>420</x>
<y>230</y>
<width>71</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel {
    color: rgb(0, 255, 0);
    font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>123</string>
</property>
</widget>
<widget class="QLabel" name="results_2">
<property name="geometry">
<rect>
<x>350</x>
<y>150</y>
<width>111</width>
<height>91</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel {
    color: rgb(255, 255, 255);
    font: 900 20pt &quot;Arial Black&quot;;
} </string>
</property>
<property name="text">
<string>Results</string>
</property>
</widget>
<widget class="QPushButton"
name="resultPageExitButton_2">
<property name="geometry">
<rect>
<x>600</x>
<y>540</y>
<width>181</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton {
    background-color: rgb(202, 26, 26);
    font-family: 'Montserrat';
    font-size: 21px;
    font-weight: bold;
    color: #ffffff;
    padding: 9px 25px;
    border-radius: 20px;
    border: 1px solid black;
    border-left: 1px solid rgb(170, 0, 0);
    border-right: 1px solid rgb(170, 0, 0);
    border-bottom: 5px solid rgb(170, 0, 0);
}

```

```

QPushButton:hover {
    background-color: rgb(247, 0, 0);
    font-family: 'Montserrat';
    font-size: 21px;
    font-weight: bold;
    color: #ffffff;

```

```

padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:pressed{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
} </string>
</property>
<property name="text">
<string>TRY AGAIN</string>
</property>
</widget>
<widget class="QLabel" name="totalAnswersResult_2">
<property name="geometry">
<rect>
<x>320</x>
<y>280</y>
<width>71</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>Total:</string>
</property>
</widget>
<widget class="QLabel" name="correctAnswersResult_2">
<property name="geometry">
<rect>
<x>320</x>
<y>230</y>
<width>81</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(0, 255, 0);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>Correct: </string>
</property>
</widget>
<widget class="QLabel" name="totalAnswersResultValue_2">
<property name="geometry">
<rect>
<x>420</x>
<y>280</y>
<width>71</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>123</string>
</property>
</widget>
</widget>
<widget class="QWidget" name="transposerResultPage">
<widget class="QLabel" name="totalAnswersResultValue_3">
<property name="geometry">
<rect>
<x>420</x>
<y>280</y>
<width>71</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>123</string>
</property>
</widget>
<widget class="QPushButton"
name="resultPageExitButton_3">
<property name="geometry">
<rect>
<x>600</x>
<y>540</y>
<width>181</width>
<height>51</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QPushButton{
background-color: rgb(202, 26, 26);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}
}

QPushButton:pressed{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

```



```

} </string>
</property>
<property name="text">
<string>TRY AGAIN</string>
</property>
</widget>
<widget class="QLabel" name="correctAnswersResult_3">
<property name="geometry">
<rect>
<x>320</x>
<y>230</y>
<width>81</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(0, 255, 0);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>Correct: </string>
</property>
</widget>
<widget class="QLabel" name="results_3">
<property name="geometry">
<rect>
<x>350</x>
<y>150</y>
<width>111</width>
<height>91</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 900 20pt &quot;Arial Black&quot;;
} </string>
</property>
<property name="text">
<string>Results</string>
</property>
</widget>
<widget class="QTextEdit" name="resultText_3">
<property name="geometry">
<rect>
<x>200</x>
<y>330</y>
<width>411</width>
<height>61</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QTextEdit{
color: rgb(255, 255, 255);
} </string>
</property>
<property name="readOnly">
<bool>true</bool>
</property>
<property name="html">
<string>&lt;!DOCTYPE HTML PUBLIC &quot;-
//W3C//DTD HTML 4.0//EN&quot;
&quot;http://www.w3.org/TR/REC-html40/strict.dtd&quot;&gt;
&lt;html&gt;&lt;head&gt;&lt;meta name=&quot;richtext&quot;
content=&quot;l&quot; /&gt;&lt;meta charset=&quot;utf-8&quot;
/&gt;&lt;style type=&quot;css&quot;&gt;
p, li { white-space: pre-wrap; }
hr { height: 1px; border-width: 0; }
li.unchecked::marker { content: &quot;\2610&quot;; }
li.checked::marker { content: &quot;\2612&quot;; }
&lt;/style&gt;&lt;/head&gt;&lt;body style=&quot;font-
family:'Segoe UI'; font-size:9pt; font-weight:400; font-
style:normal&quot;&gt;
&lt;p align=&quot;center&quot; style=&quot;margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot;font-

```

```

size: 14pt;&quot;&gt;Your transposition knowledge will improve
with each attempt, keep up the good
work.&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;&lt;/string
>
</property>
</widget>
<widget class="QLabel"
name="correctAnswersResultValue_3">
<property name="geometry">
<rect>
<x>420</x>
<y>230</y>
<width>71</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(0, 255, 0);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>123</string>
</property>
</widget>
<widget class="QLabel" name="totalAnswersResult_3">
<property name="geometry">
<rect>
<x>320</x>
<y>280</y>
<width>71</width>
<height>31</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QLabel{
color: rgb(255, 255, 255);
font: 14pt &quot;MV Boli&quot;;
} </string>
</property>
<property name="text">
<string>Total:</string>
</property>
</widget>
</widget>
<widget class="QWidget" name="transposerStartPage">
<widget class="QLabel" name="transposerHeader">
<property name="geometry">
<rect>
<x>500</x>
<y>130</y>
<width>181</width>
<height>81</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font: 900 22pt &quot;Arial Black&quot;;
color: rgb(255, 255, 255);</string>
</property>
<property name="text">
<string>Transposer</string>
</property>
</widget>
<widget class="QTextEdit" name="mainTransposerText">
<property name="geometry">
<rect>
<x>420</x>
<y>190</y>
<width>331</width>
<height>121</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QTextEdit{
color: rgb(255, 255, 255);
} </string>
</property>

```

```

        <property name="readOnly">
        <bool>true</bool>
    </property>
    <property name="html">
    <string>&lt;!DOCTYPE HTML PUBLIC &quot;-
//W3C//DTD HTML 4.0//EN&quot;
&quot;http://www.w3.org/TR/REC-html40/strict.dtd&quot;&gt;
&lt;html&gt;&lt;head&gt;&lt;meta name=&quot;qrichtext&quot;
content=&quot;1&quot; /&gt;&lt;meta charset=&quot;utf-8&quot;
/&gt;&lt;style type=&quot;text/css&quot;&gt;
p, li { white-space: pre-wrap; }
hr { height: 1px; border-width: 0; }
li.unchecked::marker { content: &quot;\2610&quot;; }
li.checked::marker { content: &quot;\2612&quot;; }
&lt;/style&gt;&lt;/head&gt;&lt;body style=&quot; font-
family:'Segoe UI'; font-size:9pt; font-weight:400; font-
style:normal;&quot;&gt;
&lt;p align=&quot;center&quot; style=&quot; margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot; font-
size:14pt;&quot;&gt;&lt;With transposer you can
freely&lt;/span&gt;&lt;/p&gt;
&lt;p align=&quot;center&quot; style=&quot; margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot; font-
size:14pt;&quot;&gt;&lt;transpose any note to needed
result&lt;/span&gt;&lt;/p&gt;
&lt;p align=&quot;center&quot; style=&quot; margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot; font-
size:14pt;&quot;&gt;&lt;note and get the
semitones&lt;/span&gt;&lt;/p&gt;
&lt;p align=&quot;center&quot; style=&quot; margin-top:0px;
margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-
indent:0; text-indent:0px;&quot;&gt;&lt;span style=&quot; font-
size:14pt;&quot;&gt;&lt;between
them.&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
    </property>
</widget>
<widget class="QPushButton" name="learnButtonT">
    <property name="geometry">
    <rect>
    <x>60</x>
    <y>500</y>
    <width>191</width>
    <height>61</height>
    </rect>
    </property>
    <property name="styleSheet">
    <string notr="true">QPushButton {
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 144, 255);
border: 1px solid black;
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);
    }

QPushButton: hover {
background-color: rgb(0, 144, 255);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);
    }

QPushButton: pressed {
background-color: rgb(0, 144, 255);
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);
border-bottom: none;
    }
    </string>
    </property>
    <property name="text">
    <string>LEARN</string>
    </property>
    </widget>
    <widget class="QLabel" name="imageT">
    <property name="geometry">
    <rect>
    <x>100</x>
    <y>130</y>
    <width>251</width>
    <height>221</height>
    </rect>
    </property>
    <property name="styleSheet">
    <string notr="true"/>
    </property>
    <property name="text">
    <string/>
    </property>
    <property name="pixmap">
    <pixmap
resource="..:/resources/Images.qrc">:/common/img_resources/trans-
poserMain.png</pixmap>
    </property>
    <property name="scaledContents">
    <bool>true</bool>
    </property>
    </widget>
    <widget class="QComboBox"
name="transpositionsSelectorComboBox">
    <property name="geometry">

```

```

    </string>
    </property>
    <property name="text">
    <string>LEARN</string>
    </property>
    </widget>
    <widget class="QPushButton" name="toolButtonT">
    <property name="geometry">
    <rect>
    <x>570</x>
    <y>500</y>
    <width>191</width>
    <height>61</height>
    </rect>
    </property>
    <property name="styleSheet">
    <string notr="true">QPushButton {
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 85, 0);
border: 1px solid black;
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
    }

QPushButton: hover {
background-color: rgb(8, 126, 0);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
    }

QPushButton: pressed {
background-color: rgb(8, 126, 0);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
border-bottom: none;
    }
    </string>
    </property>
    <property name="text">
    <string>TOOL</string>
    </property>
    </widget>
    <widget class="QLabel" name="imageT">
    <property name="geometry">
    <rect>
    <x>100</x>
    <y>130</y>
    <width>251</width>
    <height>221</height>
    </rect>
    </property>
    <property name="styleSheet">
    <string notr="true"/>
    </property>
    <property name="text">
    <string/>
    </property>
    <property name="pixmap">
    <pixmap
resource="..:/resources/Images.qrc">:/common/img_resources/trans-
poserMain.png</pixmap>
    </property>
    <property name="scaledContents">
    <bool>true</bool>
    </property>
    </widget>
    <widget class="QComboBox"
name="transpositionsSelectorComboBox">
    <property name="geometry">

```

```

    <rect>
    <x>30</x>
    <y>450</y>
    <width>261</width>
    <height>41</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover {
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
} </string>
    </property>
    <property name="currentIndex">
    <number>0</number>
    </property>
    <property name="placeholderText">
    <string>Choose transpositions amount...</string>
    </property>
    <item>
    <property name="text">
    <string>5 Transpositions</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>10 Transpositions</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>20 Transpositions</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>Endless</string>
    </property>
    </item>
  </widget>
</widget>
<widget class="QWidget" name="transposerMainPage">
<widget class="QPushButton" name="playTP">
  <property name="geometry">
    <rect>
    <x>210</x>
    <y>360</y>
    <width>151</width>
    <height>51</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton {
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 144, 255);
border: 1px solid black;
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);
}

QPushButton:hover {
background-color: rgb(0, 144, 255);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);
}

QPushButton:pressed {
background-color: rgb(0, 144, 255);
border-left: 1px solid rgb(0, 98, 255);
border-right: 1px solid rgb(0, 98, 255);
border-bottom: 5px solid rgb(0, 98, 255);
border-bottom: none;
} </string>
    </property>
    <property name="text">
    <string>Play ▶</string>
    </property>
  </widget>
<widget class="QPushButton" name="transposeButtonTP">
  <property name="geometry">
    <rect>
    <x>600</x>
    <y>550</y>
    <width>191</width>
    <height>51</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton {
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 11px 25px;
border-radius: 20px;
background-color: rgb(0, 85, 0);
border: 1px solid black;
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:hover {
background-color: rgb(8, 126, 0);
border: none;
color: rgb(226, 234, 216);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
}

QPushButton:pressed {
background-color: rgb(8, 126, 0);
border-left: 1px solid rgb(26, 65, 0);
border-right: 1px solid rgb(26, 65, 0);
border-bottom: 5px solid rgb(26, 65, 0);
border-bottom: none;
} </string>
    </property>
    <property name="text">
    <string>Transpose</string>
    </property>
  </widget>

```

```

</widget>
<widget class="QLabel" name="pianosNote">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>77</y>
      <width>751</width>
      <height>271</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QLabel {
border-radius: 5px;
border: 2px solid rgb(57, 57, 57);
} </string>
  </property>
  <property name="text">
    <string/>
  </property>
  <property name="pixmap">
    <pixmap
resource=".:resources/Images.qrc">:/common/img_resources/pian
oActivate.jif</pixmap>
  </property>
  <property name="scaledContents">
    <bool>true</bool>
  </property>
</widget>
<widget class="QComboBox" name="noteToTranspose">
  <property name="geometry">
    <rect>
      <x>310</x>
      <y>430</y>
      <width>210</width>
      <height>31</height>
    </rect>
  </property>
  <property name="minimumSize">
    <size>
      <width>210</width>
      <height>0</height>
    </size>
  </property>
  <property name="styleSheet">
    <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover {
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
} </string>
  </property>
  <property name="currentIndex">
    <number>-1</number>
  </property>
  <property name="placeholderText">

```

```

    <string>Choose a degree...</string>
  </property>
  <item>
    <property name="text">
      <string>C</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>D</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>E</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>F</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>G</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>A</string>
    </property>
  </item>
  <item>
    <property name="text">
      <string>B</string>
    </property>
  </item>
</widget>
<widget class="QComboBox" name="octaveToTranspose">
  <property name="geometry">
    <rect>
      <x>310</x>
      <y>510</y>
      <width>210</width>
      <height>31</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover {
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
} </string>
  </property>

```

```

    <property name="currentIndex">
    <number>-1</number>
    </property>
    <property name="placeholderText">
    <string>Choose octave...</string>
    </property>
    </item>
    <property name="text">
    <string>Small octave</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>1 octave</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>2 octave</string>
    </property>
    </item>
  </widget>
  <widget class="QComboBox" name="signToTranspose">
    <property name="geometry">
    <rect>
    <x>310</x>
    <y>470</y>
    <width>210</width>
    <height>31</height>
    </rect>
    </property>
    <property name="minimumSize">
    <size>
    <width>210</width>
    <height>0</height>
    </size>
    </property>
    <property name="styleSheet">
    <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover {
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
}
</string>
    </property>
    <property name="currentIndex">
    <number>-1</number>
    </property>
    <property name="placeholderText">
    <string>Choose alteration sign...</string>
    </property>
    </item>
    <property name="text">
    <string>None</string>
    </property>

```

```

    </item>
    <item>
    <property name="text">
    <string>#</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>b</string>
    </property>
    </item>
  </widget>
  <widget class="QComboBox"
name="semitonesToTranspose">
    <property name="geometry">
    <rect>
    <x>310</x>
    <y>550</y>
    <width>210</width>
    <height>41</height>
    </rect>
    </property>
    <property name="styleSheet">
    <string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover {
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
}
</string>
    </property>
    <property name="currentIndex">
    <number>-1</number>
    </property>
    <property name="placeholderText">
    <string>Choose semitones...</string>
    </property>
    </item>
    <property name="text">
    <string>-12 semitones</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>-11 semitones</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>-10 semitones</string>
    </property>
    </item>
    <item>
    <property name="text">
    <string>-9 semitones</string>
    </property>

```

```

</item>
<item>
  <property name="text">
    <string>-8 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-7 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-6 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-5 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-4 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-3 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-2 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>-1 semitone</string>
  </property>
</item>
<item>
  <property name="text">
    <string>0 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>1 semitone</string>
  </property>
</item>
<item>
  <property name="text">
    <string>2 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>3 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>4 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>5 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>6 semitones</string>
  </property>
</item>
<property name="text">

```

```

    <string>7 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>8 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>9 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>10 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>11 semitones</string>
  </property>
</item>
<item>
  <property name="text">
    <string>12 semitones</string>
  </property>
</item>
</widget>
<widget class="QSlider" name="audioVolumeSlide_3">
  <property name="geometry">
    <rect>
      <x>370</x>
      <y>370</y>
      <width>251</width>
      <height>31</height>
    </rect>
  </property>
  <property name="maximum">
    <number>100</number>
  </property>
  <property name="sliderPosition">
    <number>100</number>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
</widget>
<widget class="QPushButton" name="exitButton_3">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>550</y>
      <width>191</width>
      <height>51</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">QPushButton{
background-color: rgb(202, 26, 26);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:hover{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;

```

```

color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
}

QPushButton:pressed{
background-color: rgb(247, 0, 0);
font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: #ffffff;
padding: 9px 25px;
border-radius: 20px;
border: 1px solid black;
border-left: 1px solid rgb(170, 0, 0);
border-right: 1px solid rgb(170, 0, 0);
border-bottom: 5px solid rgb(170, 0, 0);
} </string>
</property>
<property name="text">
<string> ✖ EXIT</string>
</property>
<property name="iconSize">
<size>
<width>10</width>
<height>10</height>
</size>
</property>
</widget>
<widget class="QLabel" name="redDot1">
<property name="geometry">
<rect>
<x>33</x>
<y>310</y>
<width>31</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>
</property>
<property name="pixmap">
<pixmap
resource=" ../resources/Images.qrc">:/common/img_resources/dot1.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="redDot2">
<property name="geometry">
<rect>
<x>282</x>
<y>310</y>
<width>31</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>
</property>
<property name="pixmap">
<pixmap
resource=" ../resources/Images.qrc">:/common/img_resources/dot1.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="blueDot1">
<property name="geometry">
<rect>
<x>96</x>
<y>200</y>
<width>21</width>
<height>21</height>
</rect>
</property>
<property name="text">
<string>
</property>
<property name="pixmap">
<pixmap
resource=" ../resources/Images.qrc">:/common/img_resources/UPdot1.png</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QLabel" name="totalAnswersLabel_3">
<property name="geometry">
<rect>
<x>30</x>
<y>20</y>
<width>111</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: white;
padding: 11px 25px;
border-radius: 15px;
border: 1px solid black;
background-color: rgb(124, 124, 124);</string>
</property>
<property name="text">
<string>0/20</string>
</property>
</widget>
<widget class="QLabel" name="correctAnswersLabel_3">
<property name="geometry">
<rect>
<x>670</x>
<y>20</y>
<width>111</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">font-family: 'Montserrat';
font-size: 21px;
font-weight: bold;
color: rgb(0, 206, 0);
padding: 11px 25px;
border-radius: 15px;

```

```

border: 1px solid black;
background-color: rgb(124, 124, 124);</string>
</property>
<property name="text">
<string>0/20</string>
</property>
</widget>
<widget class="QComboBox"
name="semitonesToTranspose_Learn">
<property name="geometry">
<rect>
<x>310</x>
<y>550</y>
<width>210</width>
<height>41</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true">QComboBox {
color: rgb(255, 255, 255);
background-color: rgb(88, 88, 88);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
border: none;
}

QComboBox:hover {
background-color: rgb(144, 144, 144);
border-radius: 10px;
border-color: beige;
font: bold 14px;
min-width: 10em;
padding: 10px;
}

QComboBox:pressed {
border-style: inset;
}

QComboBox::item:hover {
background-color: rgb(170, 170, 255);
color: rgb(0, 0, 0);
}</string>
</property>
<property name="currentIndex">
<number>-1</number>
</property>
<property name="placeholderText">
<string>Choose semitones...</string>
</property>
<item>
<property name="text">
<string>0 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>1 semitone</string>
</property>
</item>
<item>
<property name="text">
<string>2 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>3 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>4 semitones</string>
</property>

```

```

</item>
<item>
<property name="text">
<string>5 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>6 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>7 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>8 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>9 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>10 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>11 semitones</string>
</property>
</item>
<item>
<property name="text">
<string>12 semitones</string>
</property>
</item>
</widget>
</widget>
<widget class="QWidget" name="aboutUsPage">
<widget class="QLabel" name="photo">
<property name="geometry">
<rect>
<x>40</x>
<y>170</y>
<width>321</width>
<height>251</height>
</rect>
</property>
<property name="text">
<string/>
</property>
<property name="pixmap">
<pixmap
resource=" ../resources/Images.qrc">:/common/img_resources/auth
ors.jpeg</pixmap>
</property>
<property name="scaledContents">
<bool>true</bool>
</property>
</widget>
<widget class="QTextEdit" name="aboutUsMainText">
<property name="geometry">
<rect>
<x>380</x>
<y>160</y>
<width>391</width>
<height>271</height>
</rect>
</property>
<property name="styleSheet">
<string notr="true"> color: rgb(255, 255, 255);</string>
</property>
<property name="readOnly">

```





```

namespace file
{
    // Images
    static const std::string volumeFile{
        "common/img_resources/volume.png" };
    static const std::string
        muteFile{"common/img_resources/mute.png"};

    // Styles
    static const std::string audioButtonStyleFile{
        "styles/styles/audioButtonStyle.qss" };
    static const std::string mutedAudioButtonStyleFile{
        "styles/styles/mutedAudioButtonStyle.qss" };
}

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent),
    uiPtr_{ std::make_shared<Ui::MainWindow>() },
    audioEnginePtr_{ std::make_shared<qt_ui::AudioEngine>() },
    NIctrlPtr_{
        std::make_unique<qt_ui::NotesIdentifierController>(uiPtr_,
            audioEnginePtr_) },
    TBctrlPtr_{
        std::make_unique<qt_ui::TonalityBuilderController>(uiPtr_,
            audioEnginePtr_) },
    TPctrlPtr_{
        std::make_unique<qt_ui::TransposerController>(uiPtr_,
            audioEnginePtr_) }
{
    uiPtr_->setUpUi(this);
    this->setFixedSize(this->geometry().width(), this-
        >geometry().height());

    // Отображение стартовой страницы (домашней страницы)
    uiPtr_->stackedWidget->setStyleSheet("background-color:
        rgba(0,0,0,0)");
    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_->homePage);

    uiPtr_->audioVolumeSlide->setTracking(true);
    uiPtr_->audioVolumeSlide_2->setTracking(true);
    uiPtr_->audioVolumeSlide_3->setTracking(true);
}

MainWindow::~~MainWindow()
{}

void MainWindow::on_IdentifyNoteButton_clicked()
{
    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_-
        >notesIdentifierStartPage);
}

void MainWindow::on_HomeButton_clicked()
{
    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_->homePage);
}

void MainWindow::on_beginButtonNI_clicked()
{
    NIctrlPtr_->BeginTheFlow();
}

void MainWindow::on_nextButton_clicked()
{
    NIctrlPtr_->ProceedAnswer();
}

void MainWindow::on_exitButton_clicked()
{
    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_-
        >notesIdentifierStartPage);
}

void MainWindow::on_soundControl_clicked()

```

```

{
    // TODO: возможно можно выделить в один метод
    QFile stylesFile;
    const bool currentState{ NIctrlPtr_->IsSoundOn() };
    if(currentSoundState)
    {
        // на серую, иконка мута

        stylesFile.setFileName(QString::fromStdString(fmt::format("/{}/",
            file::mutedAudioButtonStyleFile)));
        uiPtr_->soundControl-
            >setIcon(QIcon(QString::fromStdString(fmt::format("/{}/",
                file::muteFile))));
    }
    else
    {
        // на зеленую, иконка обычная

        stylesFile.setFileName(QString::fromStdString(fmt::format("/{}/",
            file::audioButtonStyleFile)));
        uiPtr_->soundControl-
            >setIcon(QIcon(QString::fromStdString(fmt::format("/{}/",
                file::volumeFile))));
    }

    NIctrlPtr_->SetSoundState(!currentState);

    stylesFile.open(QFile::ReadOnly);
    uiPtr_->soundControl-
        >setStyleSheet(QLatin1String(stylesFile.readAll()));
    uiPtr_->soundControl->repaint();
}

void MainWindow::on_resultPageExitButton_clicked()
{
    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_-
        >notesIdentifierStartPage);
}

void MainWindow::on_toolButtonTB_clicked()
{
    uiPtr_->modeSelectorTB->setCurrentIndex(-1);
    uiPtr_->tonicSelectorTB->setCurrentIndex(-1);
    uiPtr_->signSelectorTB->setCurrentIndex(-1);

    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_-
        >tonalityBuilderMainPage);
    TBctrlPtr_->BeginToolFlow();
}

void MainWindow::on_learnButtonTB_clicked()
{
    uiPtr_->modeSelectorTB->setCurrentIndex(-1);
    uiPtr_->tonicSelectorTB->setCurrentIndex(-1);
    uiPtr_->signSelectorTB->setCurrentIndex(-1);

    uiPtr_->stackedWidget->setCurrentWidget(uiPtr_-
        >tonalityBuilderMainPage);
    TBctrlPtr_->BeginLearnFlow();
}

void MainWindow::on_playTB_clicked()
{
    try
    {
        TBctrlPtr_->PlayBuiltTonality();
    }
    catch (const std::exception& e)
    {
        QMessageBox::critical(this, "Error", e.what());
    }
    catch (...)
    {
    }
}

```

```

        QMessageBox::critical(this, "Error", "Unknown error
occured!");
    }
}

```

```

void MainWindow::on_buildTB_clicked()
{
    try
    {
        if (TBctrlPtr->IsToolMode())
        {
            TBctrlPtr->BuildTonality();
        }
        else
        {
            TBctrlPtr->ProceedAnswer();
        }
    }
    catch (const std::exception& e)
    {
        QMessageBox::critical(this, "Error", e.what());
    }
    catch (...)
    {
        QMessageBox::critical(this, "Error", "Unknown error
occured!");
    }
}

```

```

void MainWindow::on_resultPageExitButton_2_clicked()
{
    uiPtr->stackedWidget->setCurrentWidget(uiPtr->
tonalityBuilderStartPage);
}

```

```

void MainWindow::on_TonalityBuilderButton_clicked()
{
    uiPtr->stackedWidget->setCurrentWidget(uiPtr->
tonalityBuilderStartPage);

    uiPtr->tonalitiesSelectorComboBox->setCurrentIndex(0);
}

```

```

void MainWindow::on_tonicSelectorTB_currentIndexChanged(int
index)
{
    if (index != -1 && uiPtr->signSelectorTB->currentIndex() != -
1 && uiPtr->modeSelectorTB->currentIndex() != -1)
    {
        TBctrlPtr->ControlBuildNextButtonState(true);
    }
}

```

```

void MainWindow::on_signSelectorTB_currentIndexChanged(int
index)
{
    if (index != -1 && uiPtr->tonicSelectorTB->currentIndex() != -
1 && uiPtr->modeSelectorTB->currentIndex() != -1)
    {
        TBctrlPtr->ControlBuildNextButtonState(true);
    }
}

```

```

void MainWindow::on_modeSelectorTB_currentIndexChanged(int
index)
{
    if (index != -1 && uiPtr->tonicSelectorTB->currentIndex() != -
1 && uiPtr->signSelectorTB->currentIndex() != -1)
    {
        TBctrlPtr->ControlBuildNextButtonState(true);
    }
}

```

```

void MainWindow::on_exitButton_2_clicked()
{
    uiPtr->stackedWidget->setCurrentWidget(uiPtr->
tonalityBuilderStartPage);
}

```

```

void MainWindow::on_stackedWidget_currentChanged(int arg1)
{
    // Любые звуки при смене окон нужно отключать.
    audioEnginePtr->StopAudio();
}

```

```

void MainWindow::on_transposeButtonTP_clicked()
{
    try
    {
        if (TPctrlPtr->IsToolMode())
        {
            TPctrlPtr->Transpose();
        }
        else
        {
            TPctrlPtr->ProceedAnswer();
        }
    }
    catch (const std::exception& e)
    {
        QMessageBox::critical(this, "Error", e.what());
    }
    catch (...)
    {
        QMessageBox::critical(this, "Error", "Unknown error
occured!");
    }
}

```

```

void MainWindow::on_playTP_clicked()
{
    try
    {
        TPctrlPtr->PlayTransposition();
    }
    catch (const std::exception& e)
    {
        QMessageBox::critical(this, "Error", e.what());
    }
    catch (...)
    {
        QMessageBox::critical(this, "Error", "Unknown error
occured!");
    }
}

```

```

void MainWindow::on_exitButton_3_clicked()
{
    uiPtr->stackedWidget->setCurrentWidget(uiPtr->
transposerStartPage);
}

```

```

void
MainWindow::on_noteToTranspose_currentIndexChanged(int
index)
{
    if (index != -1 && uiPtr->signToTranspose->currentIndex() !=
-1 &&
        uiPtr->octaveToTranspose->currentIndex() != -1 &&
        uiPtr->semitonesToTranspose->currentIndex() != -1)
    {
        TPctrlPtr->ControlTransposeNextButtonState(true);
    }
}

```

```

void
MainWindow::on_signToTranspose_currentIndexChanged(int
index)
{
    if (index != -1 && uiPtr_ ->noteToTranspose->currentIndex() !=
-1 &&
        uiPtr_ ->octaveToTranspose->currentIndex() != -1 &&
        uiPtr_ ->semitonesToTranspose->currentIndex() != -1)
    {
        TPctrlPtr_ ->ControlTransposeNextButtonState(true);
    }
}

```

```

void
MainWindow::on_octaveToTranspose_currentIndexChanged(int
index)
{
    if (index != -1 && uiPtr_ ->noteToTranspose->currentIndex() !=
-1 &&
        uiPtr_ ->signToTranspose->currentIndex() != -1 &&
        uiPtr_ ->semitonesToTranspose->currentIndex() != -1)
    {
        TPctrlPtr_ ->ControlTransposeNextButtonState(true);
    }
}

```

```

void
MainWindow::on_semitonesToTranspose_currentIndexChanged(i
nt index)
{
    if (index != -1 && uiPtr_ ->signToTranspose->currentIndex() !=
-1 &&
        uiPtr_ ->octaveToTranspose->currentIndex() != -1 &&
        uiPtr_ ->noteToTranspose->currentIndex() != -1)
    {
        TPctrlPtr_ ->ControlTransposeNextButtonState(true);
    }
}

```

```

void MainWindow::on_TransposerButton_clicked()
{
    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>transposerStartPage);
}

```

```

void MainWindow::on_resultPageExitButton_3_clicked()
{
    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>transposerStartPage);
}

```

```

void MainWindow::on_toolButtonT_clicked()
{
    uiPtr_ ->noteToTranspose->setCurrentIndex(-1);
    uiPtr_ ->signToTranspose->setCurrentIndex(-1);
    uiPtr_ ->octaveToTranspose->setCurrentIndex(-1);
    uiPtr_ ->semitonesToTranspose->setCurrentIndex(-1);

    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>transposerMainPage);
    TPctrlPtr_ ->BeginToolFlow();
}

```

```

void MainWindow::on_learnButtonT_clicked()
{
    uiPtr_ ->noteToTranspose->setCurrentIndex(-1);
    uiPtr_ ->signToTranspose->setCurrentIndex(-1);
    uiPtr_ ->octaveToTranspose->setCurrentIndex(-1);
    uiPtr_ ->semitonesToTranspose->setCurrentIndex(-1);
}

```

```

    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>transposerMainPage);
    TPctrlPtr_ ->BeginLearnFlow();
}

```

```

void
MainWindow::on_semitonesToTranspose_Learn_currentIndexCha
nged(int index)
{
    if (index != -1)
    {
        TPctrlPtr_ ->ControlTransposeNextButtonState(true);
    }
}

```

```

void MainWindow::on_aboutUsButton_clicked()
{
    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>aboutUsPage);
}

```

## mainwindow.h:

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

```

```

#include <QMainWindow>

```

```

namespace qt_ui
{
    class AudioEngine;
    class NotesIdentifierController;
    class TonalityBuilderController;
    class TransposerController;
}

```

```

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

```

```

class MainWindow : public QMainWindow
{
    Q_OBJECT

```

```

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

```

```

private slots:
    void on_beginButtonNI_clicked();
    void on_HomeButton_clicked();
    void on_IdentifyNoteButton_clicked();
    void on_nextButton_clicked();
    void on_exitButton_clicked();
    void on_soundControl_clicked();
    void on_resultPageExitButton_clicked();

```

```

    void on_toolButtonTB_clicked();

```

```

    void on_learnButtonTB_clicked();

```

```

    void on_playTB_clicked();

```

```

    void on_buildTB_clicked();

```

```

    void on_resultPageExitButton_2_clicked();

```

```

    void on_TonalityBuilderButton_clicked();

```

```

    void on_tonicSelectorTB_currentIndexChanged(int index);

```

```

    void on_signSelectorTB_currentIndexChanged(int index);

```

```

void on_modeSelectorTB_currentIndexChanged(int index);

void on_exitButton_2_clicked();

void on_stackedWidget_currentChanged(int arg1);

void on_transposeButtonTP_clicked();

void on_playTP_clicked();

void on_exitButton_3_clicked();

void on_noteToTranspose_currentIndexChanged(int index);

void on_signToTranspose_currentIndexChanged(int index);

void on_octaveToTranspose_currentIndexChanged(int index);

void on_semitonesToTranspose_currentIndexChanged(int
index);

void on_TransposerButton_clicked();

void on_resultPageExitButton_3_clicked();

void on_toolButtonT_clicked();

void on_learnButtonT_clicked();

void
on_semitonesToTranspose_Learn_currentIndexChanged(int
index);

void on_aboutUsButton_clicked();

private:
    std::shared_ptr<Ui::MainWindow> uiPtr_;
    std::shared_ptr<qt_ui::AudioEngine> audioEnginePtr_;
    std::unique_ptr<qt_ui::NotesIdentifierController> NIctrlPtr_;
    std::unique_ptr<qt_ui::TonalityBuilderController> TBctrlPtr_;
    std::unique_ptr<qt_ui::TransposerController> TPctrlPtr_;
    // Если нужно добавить слот, и дизайнер не дает это сделать
-
    // РАСкомменти эту строку VVV
    // Ui::MainWindow *Ui;
};
#endif // MAINWINDOW_H

```

## tonalitybuilderController.h:

```

#ifndef TONALITYBUILDERCONTROLLER_H
#define TONALITYBUILDERCONTROLLER_H

#include <QMainWindow>

#include <backend/data/include/Types.h>
#include <backend/data/include/Scale.h>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

namespace qt_ui
{
class AudioEngine;

class TonalityBuilderController
{
    //
    // Constructors
    //
public:
    TonalityBuilderController(std::shared_ptr<Ui::MainWindow>
uiPtr, std::shared_ptr<qt_ui::AudioEngine> audioEnginePtr);

```

```

//
// Public interface.
//
public:
    //! Инициализирует начало работы тренажера.
    void BeginLearnFlow();
    //! Инициализирует начало работы инструментария.
    void BeginToolFlow();
    //! Активирует/деактивирует кнопку Build/Next, в т.ч.
изменяя дизайн.
    void ControlBuildNextButtonState(const bool enable) const;
    //! Возвращает режим.
    bool IsToolMode() const;
    //! Обрабатывает ответ, данный пользователем.
    void ProceedAnswer();
    //! Строит тональность.
    void BuildTonality();
    //! Проигрывает текущую тональность.
    void PlayBuiltTonality() const;

//
// Private methods.
//
private:
    //! Активирует/деактивирует кнопку Play, в т.ч. изменяя
дизайн.
    void ControlPlayButtonState(const bool enable) const;
    //! Устанавливает позиции точек в соответствии с
тональностью.
    void SetTonalityDotsOnKeys(const data::Scale& scale) const;
    //! Прячет точки на клавишах пианино.
    void HideDotsOnKeys() const;
    //! Прячет/показывает элементы для обучения.
    void HideShowLearningModeElements(const bool show) const;
    //! Считывает значение максимального количества
тональностей для угадывания.
    std::uint8_t SelectedIndexToTonalityAmount(const uint8_t
index) const;
    //! Корректирует тональность, меняя октавы.
    void ValidateTonality();

//
// Private members.
//
private:
    std::shared_ptr<Ui::MainWindow> uiPtr_;
    std::shared_ptr<qt_ui::AudioEngine> audioEnginePtr_;
    //! Текущая построенная тональность.
    data::Scale currentTonality_;
    //! Бесконечный режим или нет.
    bool isEndless_ = false;
    //! Режим инструмента или нет.
    bool isToolMode_;
    //! Выбранное количество тональностей.
    std::uint8_t maxAmount_;
};

}

#endif // TONALITYBUILDERCONTROLLER_H

```

## tonalitybuildercontroller.cpp:

```

// Если нужно добавить слот, и дизайнер не дает это сделать -
// ЗАкомменти этот инклюд VVV
#include "ui_mainwindow.h"

```

```

#include "../src/pch.h"

#include <QDebug>
#include <QFile>
#include <QMap>
#include <QMultiMap>
#include <QPixmap>
#include <QPoint>
#include <QString>
#include <QTimer>

#include "../include/AudioEngine.h"
#include "../include/TonalityBuilderController.h"

#include <backend/elegia/include/ScaleBuilder.h>

namespace key
{

static const std::string NoneSign{ "None" };

} // namespace key

namespace details
{

static const std::string NextButtonText{ "Next ►" };
static const std::string BuildButtonText{ "Build" };

// Styles
static const std::string buildTBButtonStyleFile{
"styles/styles/buildTBButtonStyle.qss" };
static const std::string playTBButtonStyleFile{
"styles/styles/playTBButtonStyle.qss" };
static const std::string disabledBuildTBButtonStyleFile{
"styles/styles/disabledBuildTBButtonStyle.qss" };
static const std::string disabledPlayTBButtonStyleFile{
"styles/styles/disabledPlayTBButtonStyle.qss" };

} //namespace details

namespace qt_ui
{

TonalityBuilderController::TonalityBuilderController(
    std::shared_ptr<Ui::MainWindow> uiPtr,
    std::shared_ptr<qt_ui::AudioEngine> audioEnginePtr ) :
    uiPtr_{ uiPtr },
    audioEnginePtr_{ audioEnginePtr }
{

}

void TonalityBuilderController::BeginLearnFlow()
{
    isToolMode_ = false;
    ControlPlayButtonState(true);
    ControlBuildNextButtonState(false);
    HideDotsOnKeys();

    try
    {
        currentTonality_ =
elegia::ScaleBuilder::GenerateTonality(true);
    }
    catch (const std::exception &e)
    {
        qCritical() << QString::fromStdString(fmt::format("Failed to
begin the Tonality Builder flow: {}", e.what()));
        throw std::runtime_error{ "Failed to process the Tonality
Builder" };
    }

    maxAmount_ = SelectedIndexToTonalityAmount(uiPtr_ -
>tonalitiesSelectorComboBox->currentIndex());
    if (maxAmount_ == 0)
    {
        isEndless_ = true;

        uiPtr_ ->totalAnswersLabel_ 2-
>setText(QString::fromStdString(std::to_string(elegia::ScaleBuilde
r::GetScalesGenerated())));
        uiPtr_ ->correctAnswersLabel_ 2-
>setText(QString::fromStdString("0"));
    }
    else
    {
        const std::string totalNotesText{fmt::format("{}{}",
elegia::ScaleBuilder::GetScalesGenerated(), maxAmount_)};
        const std::string correctNotesText{fmt::format("{}{}", 0,
maxAmount_)};

        uiPtr_ ->totalAnswersLabel_ 2-
>setText(QString::fromStdString(totalNotesText));
        uiPtr_ ->correctAnswersLabel_ 2-
>setText(QString::fromStdString(correctNotesText));
    }

    ValidateTonality();
    SetTonalityDotsOnKeys(currentTonality_);
    HideShowLearningModeElements(true);
    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>tonalityBuilderMainPage);
}

void TonalityBuilderController::BeginToolFlow()
{
    isToolMode_ = true;
    ControlPlayButtonState(false);
    ControlBuildNextButtonState(false);
    HideShowLearningModeElements(false);
    HideDotsOnKeys();

    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>tonalityBuilderMainPage);
}

bool TonalityBuilderController::IsToolMode() const
{
    return isToolMode_;
}

void TonalityBuilderController::ProceedAnswer()
{
    // Получаем ответ пользователя
    const std::string answerModeStr{ uiPtr_ ->modeSelectorTB-
>currentText().toStdString() };
    const std::string answerTonicStr{ uiPtr_ ->tonicSelectorTB-
>currentText().toStdString() };
    std::string answerSignStr{ uiPtr_ ->signSelectorTB-
>currentText().toStdString() };
    if (answerSignStr == key::NoneSign)
    {
        answerSignStr = "";
    }

    const data::ScaleRequestData scaleRequestData {
data::String2Degree(answerTonicStr),
    data::String2AlterationSign(answerSignStr),
    data::String2Key(answerModeStr)
    };

    const bool isCorrect{
elegia::ScaleBuilder::IsCorrectAnswer(scaleRequestData) };
    if (isCorrect)
    {
        const std::uint8_t correctAnswers{
elegia::ScaleBuilder::GetCorrectAnswers() };

        if (isEndless_)
        {
            uiPtr_ ->correctAnswersLabel_ 2-
>setText(QString::fromStdString(std::to_string(correctAnswers)));
        }
        else
        {

```

```

        uiPtr_ ->correctAnswersLabel_2-
>setText(QString::fromStdString(fmt::format("{} / {}",
correctAnswers, maxAmount_));
    }
}

if (elegia::ScaleBuilder::GetScalesGenerated() != maxAmount_)
{
    // Сгенерировать следующую тональность...
    try
    {
        currentTonality_ =
elegia::ScaleBuilder::GenerateTonality(false);
    }
    catch (const std::exception &e)
    {
        qCritical() << QString::fromStdString(fmt::format("Failed
to begin the Tonality Builder flow: {}", e.what()));
        throw std::runtime_error{ "Failed to process the Tonality
Builder" };
    }

    const std::uint8_t totalAnswers{
elegia::ScaleBuilder::GetScalesGenerated() };
    if (isEndless_)
    {
        uiPtr_ ->totalAnswersLabel_2-
>setText(QString::fromStdString(std::to_string(totalAnswers)));
    }
    else
    {
        uiPtr_ ->totalAnswersLabel_2-
>setText(QString::fromStdString(fmt::format("{} / {}",
totalAnswers, maxAmount_));
    }

    ValidateTonality();
    SetTonalityDotsOnKeys(currentTonality_);
}
else
{
    uiPtr_ ->totalAnswersResultValue_2-
>setText(QString::fromStdString(std::to_string(elegia::ScaleBuilde
r::GetScalesGenerated())));
    uiPtr_ ->correctAnswersResultValue_2-
>setText(QString::fromStdString(std::to_string(elegia::ScaleBuilde
r::GetCorrectAnswers())));

    uiPtr_ ->stackedWidget->setCurrentWidget(uiPtr_ -
>tonalityBuilderResultPage);
}

    ControlPlayButtonState(true);
}

void TonalityBuilderController::BuildTonality()
{
    // Получаем ответ пользователя
    const std::string answerModeStr{ uiPtr_ ->modeSelectorTB-
>currentText().toStdString() };
    const std::string answerTonicStr{ uiPtr_ ->tonicSelectorTB-
>currentText().toStdString() };
    std::string answerSignStr{ uiPtr_ ->signSelectorTB-
>currentText().toStdString() };
    if (answerSignStr == key::NoneSign)
    {
        answerSignStr = "";
    }

    const data::ScaleRequestData scaleRequestData{
data::String2Degree(answerTonicStr),
data::String2AlterationSign(answerSignStr),
data::String2Key(answerModeStr)
};
    currentTonality_ =
elegia::ScaleBuilder::BuildTonality(scaleRequestData);
    ValidateTonality();

```

```

    SetTonalityDotsOnKeys(currentTonality_);
    ControlPlayButtonState(true);
}

void TonalityBuilderController::PlayBuiltTonality() const
{
    audioEnginePtr_ ->StartAndPlayQueue(currentTonality_.notes_,
uiPtr_ ->audioVolumeSlide_2->value() / 100.0);
}

void TonalityBuilderController::ControlPlayButtonState(const
bool enable) const
{
    QFile stylesFile;
    if (!enable)
    {
        uiPtr_ ->playTB->setDisabled(true);

        // изменить стиль на серую кнопку...

        stylesFile.setFileName(QString::fromStdString(fmt::format("/{}",
details::disabledPlayTBButtonStyleFile)));
    }
    else
    {
        uiPtr_ ->playTB->setEnabled(true);

        // изменить стиль на обычную кнопку...

        stylesFile.setFileName(QString::fromStdString(fmt::format("/{}",
details::playTBButtonStyleFile)));
    }

    stylesFile.open(QFile::ReadOnly);
    uiPtr_ ->playTB-
>setStyleSheet(QLatin1String(stylesFile.readAll()));
    uiPtr_ ->playTB->repaint();
}

void
TonalityBuilderController::ControlBuildNextButtonState(const
bool enable) const
{
    QFile stylesFile;
    if (!enable)
    {
        uiPtr_ ->buildTB->setDisabled(true);

        // изменить стиль на серую кнопку...

        stylesFile.setFileName(QString::fromStdString(fmt::format("/{}",
details::disabledBuildTBButtonStyleFile)));
    }
    else
    {
        uiPtr_ ->buildTB->setEnabled(true);

        // изменить стиль на обычную кнопку...

        stylesFile.setFileName(QString::fromStdString(fmt::format("/{}",
details::buildTBButtonStyleFile)));
    }

    stylesFile.open(QFile::ReadOnly);
    uiPtr_ ->buildTB-
>setStyleSheet(QLatin1String(stylesFile.readAll()));
    uiPtr_ ->buildTB->repaint();
}

void TonalityBuilderController::SetTonalityDotsOnKeys(const
data::Scale& scale) const
{
    // Мапа расположена тут т.к. доступ к лейблам приватный.
    static const QMap<data::Note, QLabel*> NoteDotsMap{
        {data::Note{data::Degree::B, data::AlterationSign::SHARP,
data::Octave::GREAT}, uiPtr_ ->C3 },
        {data::Note{data::Degree::C, data::AlterationSign::NONE,
data::Octave::SMALL}, uiPtr_ ->C3 },

```





```

        {data::Note{data::Degree::G, data::AlterationSign::SHARP,
data::Octave::LINE2}, uiPtr_>Gs5 },
        {data::Note{data::Degree::A, data::AlterationSign::FLAT,
data::Octave::LINE2}, uiPtr_>Gs5 },

        {data::Note{data::Degree::A, data::AlterationSign::NONE,
data::Octave::LINE2}, uiPtr_>A5 },
        {data::Note{data::Degree::B,
data::AlterationSign::DOUBLE_FLAT, data::Octave::LINE2},
uiPtr_>A4 },

        {data::Note{data::Degree::A, data::AlterationSign::SHARP,
data::Octave::LINE2}, uiPtr_>As5 },
        {data::Note{data::Degree::B, data::AlterationSign::FLAT,
data::Octave::LINE2}, uiPtr_>As5 },

        {data::Note{data::Degree::B, data::AlterationSign::NONE,
data::Octave::LINE2}, uiPtr_>B5 },
        {data::Note{data::Degree::C, data::AlterationSign::FLAT,
data::Octave::LINE3}, uiPtr_>B5 },
    };

    // Прячем предыдущие точки.
    HideDotsOnKeys();
    for (const auto note : scale.notes_)
    {
        //! TODO: ситуация, схожая с бекендовской по поиску в
        // карте.
        for (auto it = NoteDotsMap.begin(); it != NoteDotsMap.end();
        ++it)
        {
            if (note == it.key())
            {
                it.value()->show();
            }
        }
    }
}

void TonalityBuilderController::HideDotsOnKeys() const
{
    // TODO: возможно в лучшем случае стоило бы применять
    // вектор указателей QVector<QLabel*>...
    uiPtr_>C3->hide();
    uiPtr_>Cs3->hide();
    uiPtr_>D3->hide();
    uiPtr_>Ds3->hide();
    uiPtr_>E3->hide();
    uiPtr_>F3->hide();
    uiPtr_>Fs3->hide();
    uiPtr_>G3->hide();
    uiPtr_>Gs3->hide();
    uiPtr_>A3->hide();
    uiPtr_>As3->hide();
    uiPtr_>B3->hide();

    uiPtr_>C4->hide();
    uiPtr_>Cs4->hide();
    uiPtr_>D4->hide();
    uiPtr_>Ds4->hide();
    uiPtr_>E4->hide();
    uiPtr_>F4->hide();
    uiPtr_>Fs4->hide();
    uiPtr_>G4->hide();
    uiPtr_>Gs4->hide();
    uiPtr_>A4->hide();
    uiPtr_>As4->hide();
    uiPtr_>B4->hide();

    uiPtr_>C5->hide();
    uiPtr_>Cs5->hide();
    uiPtr_>D5->hide();
    uiPtr_>Ds5->hide();
    uiPtr_>E5->hide();
    uiPtr_>F5->hide();
    uiPtr_>Fs5->hide();
    uiPtr_>G5->hide();
    uiPtr_>Gs5->hide();

    uiPtr_>A5->hide();
    uiPtr_>As5->hide();
    uiPtr_>B5->hide();
}

void
TonalityBuilderController::HideShowLearningModeElements(const
bool show) const
{
    if (show)
    {
        // показать счетчики (зеленый и белый), а так же поменять
        // надпись "Build" на "Next ->"
        uiPtr_>totalAnswersLabel_2->show();
        uiPtr_>correctAnswersLabel_2->show();

        // Learning mode -> "NEXT >"
        uiPtr_>buildTB-
>setText(QString::fromStdString(details::NextButtonText));
    }
    else
    {
        uiPtr_>totalAnswersLabel_2->hide();
        uiPtr_>correctAnswersLabel_2->hide();

        // Tool mode -> "BUILD"
        uiPtr_>buildTB-
>setText(QString::fromStdString(details::BuildButtonText));
    }
}

std::uint8_t
TonalityBuilderController::SelectedIndexToTonalityAmount(const
uint8_t index) const
{
    if (index == 0)
    {
        return 5;
    }
    if (index == 1)
    {
        return 10;
    }
    if (index == 2)
    {
        return 20;
    }
    if (index == 3)
    {
        return 0;
    }

    throw std::runtime_error("No acceptable index is found for
Tonality Builder");
}

void TonalityBuilderController::ValidateTonality()
{
    const auto& currTonalityNote{ currentTonality_.notes_.at(0) };

    try
    {
        if (currTonalityNote.degree_ != data::Degree::C)
        {
            for (auto& note : currentTonality_.notes_)
            {
                // TODO: D# и D bug????
                if (note < currTonalityNote)
                {
                    note.octave_ = data::Octave::LINE2;
                }
                // Кейс для одинаковых нот (первая и последняя, как
                // например в D# мажор)
                else if (note.degree_ == currTonalityNote.degree_ &&
                !(note == currTonalityNote))
                {
                    // Если последняя нота в итерации на полутон-тон
                    // выше первой - она не должна идти по кругу на пианино.

```

```

        if (note.alterationSign_ ==
data::AlterationSign::SHARP ||
            note.alterationSign_ ==
data::AlterationSign::DOUBLE_SHARP)
        {
            note.octave_ = data::Octave::LINE2;
        }
    }
}
catch (const std::exception& e)
{
    throw std::runtime_error( fmt::format("Could not validate
tonality: {}", e.what()) );
}
}
} // namespace qt_ui

```

## transposer.cpp:

```

#include "pch.h"
#include <Note.h>
#include "Transposer.h"
#include <Random.h>

namespace details
{
    static const std::uint8_t SemtonesGenerationBound{ 12 };
}

namespace elegia
{
    std::pair<data::Note, data::Note>
Transposer::GenerateTransposition(const bool isJustBegannd)
{
    // Срабатывает единоразово - при начале
использования тренажера транспонирования.
    // Каждая попытка начать заново также должна
обнулять попытки.
    if (isJustBegannd)
    {
        NullifyData();
    }

    //! TODO: повторяется и в ScaleBuilder. Что насчет
расположения этого вектора в одном общем
месте?
    static const std::vector<data::Degree> degreesVec{
        data::Degree::C, data::Degree::D,
data::Degree::E, data::Degree::F,
        data::Degree::G, data::Degree::A,
data::Degree::B,
    };

    std::uint8_t generatedSignIndex =
utils::Random::RandomizeInt32(0, 2);
    //! TODO: генерация подвязана под фронт,
возможно стоит поменять.
    std::uint8_t generatedOctaveIndex =
utils::Random::RandomizeInt32(3, 5);

    data::Note note1, note2;
    note1.degree_ =
degreesVec[utils::Random::RandomizeInt32(0, degreesVec.size() -
1)];
    note1.alterationSign_ = data::AlterationSign{
generatedSignIndex };
    note1.octave_ = data::Octave{ generatedOctaveIndex };
}

```

```

bool isSuccessful{ false };
do
{
    // Генерация интервала от 0 полутонов до
12 полутонов (оба включительно).
    generatedRange_ =
utils::Random::RandomizeInt32(0,
details::SemtonesGenerationBound);
    note2 = TransposeNote(note1,
generatedRange_);

    //! TODO: генерация подвязана под
фронт, возможно стоит поменять.
    if (note2.octave_ >= data::Octave::SMALL
&& note2.octave_ <= data::Octave::LINE2)
    {
        isSuccessful = true;
    }
} while (!isSuccessful);

transpositionsGenerated_++;

return std::pair<data::Note, data::Note>{ note1, note2
};
}

bool Transposer::IsCorrectAnswer(const std::int8_t semitones)
{
    if (semitones == generatedRange_)
    {
        correctAnswers_++;

        return true;
    }

    return false;
}

data::Note Transposer::TransposeNote(const data::Note& note,
const std::int8_t semitones)
{
    data::Note resultNote{ note };

    // Поднять
    if (semitones > 0)
    {
        for (int i = 0; i < semitones; i++)
        {
            ++resultNote;
        }
    }
    // Опустить
    else if (semitones < 0)
    {
        for (int i = 0; i > semitones; i--)
        {
            --resultNote;
        }
    }

    return resultNote;
}

std::uint8_t Transposer::GetCorrectAnswers()
{
    return correctAnswers_;
}

std::uint8_t Transposer::GetTranspositionsGenerated()
{
    return transpositionsGenerated_;
}

void Transposer::NullifyData()
{
    correctAnswers_ = 0;
    generatedRange_ = 0;
    transpositionsGenerated_ = 0;
}

```

```

}

} // namespace elegia

transposer.h:

#pragma once
#include <Types.h>

namespace elegia
{
//
// Класс для работы с транспонировщиком.
//

class Transposer
{
//
// Public interface.
//

public:
    // Генерирует случайную транспозицию двух нот.
    static std::pair<data::Note, data::Note>
    GenerateTransposition(const bool isJustBegannd);
    // Обрабатывает ответ пользователя по
    транспозиции (кол-во полутонов).
    static bool IsCorrectAnswer(const std::int8_t
    semitones);
    // Транспонирует входящую ноту на заданное
    количество полутонов.
    static data::Note TransposeNote(const data::Note&
    note, const std::int8_t semitones);
    // Возвращает количество верных ответов.
    static std::uint8_t GetCorrectAnswers();
    // Возвращает количество генераций.
    static std::uint8_t GetTranspositionsGenerated();

//
// Private methods.
//

private:
    // Обнуляет данные по угадыванию.
    static void NullifyData();

//
// Private members.
//

private:
    // Количество верно угаданных транспозиций.
    inline static std::uint8_t correctAnswers_;
    // Текущий сгенерированный интервал
    транспозиции.
    inline static std::int8_t generatedRange_;
    // Текущее количество сгенерированных
    транспозиций.
    inline static std::uint8_t transpositionsGenerated_;
};

} // namespace elegia

```

## elegiatranposertest.cpp:

```

#include "pch.h"

#include <../data/include/Types.h>
#include <../data/include/Note.h>
#include <../data/include/Scale.h>

#include <../elegia/include/Transposer.h>

/**
 * Описание: транспозиция ноты на заданный интервал.
 * Когда: транспонируется нота
 * Тогда: функция возвращает верную транспонированную
    ноту.
 */

```

```

TEST(ElegiaTransposerTest, TransposeNote)
{
    data::Note note { data::Degree::C, data::AlterationSign::NONE,
    data::Octave::LINE1 };
    std::int8_t transposeRange1 { 14 };
    std::int8_t transposeRange2 { -14 };

    data::Note expectedNote1 { data::Degree::D,
    data::AlterationSign::NONE, data::Octave::LINE2 };
    data::Note expectedNote2 { data::Degree::B,
    data::AlterationSign::FLAT, data::Octave::GREAT };

    ASSERT_EQ(elegia::Transposer::TransposeNote(note,
    transposeRange1), expectedNote1);
    ASSERT_EQ(elegia::Transposer::TransposeNote(note,
    transposeRange2), expectedNote2);
}

/**
 * Описание: генерация транспозиции и ее угадывание с
    верным ответом
 * Когда: сгенерированный интервал совпадает с
    предложенным пользователем ответом
 * Тогда: функция проверки возвращает true
 */
TEST(ElegiaTransposerTest, CorrectRange)
{
    // Возвращает пару нот.
    const auto generatedTransposition {
    elegia::Transposer::GenerateTransposition(true) };

    data::Note currentNote { generatedTransposition.first };
    std::int8_t semitones = 0;

    // Если вторая нота выше первой - интервал
    положительный.
    if (generatedTransposition.first <
    generatedTransposition.second)
    {
        do
        {
            ++currentNote;
            ++semitones;
        } while (!(currentNote ==
    generatedTransposition.second));

        // Если вторая нота ниже первой - интервал
        отрицательный.
        else if (generatedTransposition.first >
    generatedTransposition.second)
        {
            do
            {
                --currentNote;
                --semitones;
            } while (!(currentNote ==
    generatedTransposition.second));
        }

        ASSERT_TRUE(elegia::Transposer::IsCorrectAnswer(
    semitones));
    }

    /**
     * Описание: генерация транспозиции и ее неугадывание
     * Когда: сгенерированный интервал не совпадает с
        предложенным пользователем ответом
     * Тогда: функция проверки возвращает false
     */
    TEST(ElegiaTransposerTest, WrongRange)
    {
        // Возвращает пару нот.
        const auto generatedTransposition {
        elegia::Transposer::GenerateTransposition(true) };

        data::Note currentNote { generatedTransposition.first };
        std::int8_t semitones = 0;
    }
}

```

```

// Подбор ложного значения.
// Если интервала нет - задаем ответ как 1. Иначе 0.
    if (generatedTransposition.first ==
generatedTransposition.second)
    {
        semitones = 1;
    }

    ASSERT_FALSE(elegia::Transposer::IsCorrectAnswer
(semitones));
}

```

## notates.cpp:

```

#include "pch.h"

#include <../data/include/Note.h>

/**
 * Описание: инкрементация структуры Note
 * Когда: вызвана инкрементация оператор++
 * Тогда: нота инкрементирована корректно
 */
TEST(NoteTest, Incrementation)
{
    {
        data::Note note{ data::Degree::C,
data::AlterationSign::NONE, data::Octave::LINE1 };
        data::Note expectedNote{ data::Degree::C,

        data::AlterationSign::SHARP,

        data::Octave::LINE1 };
        ++note;

        ASSERT_EQ(note, expectedNote);
    }

    {
        data::Note note{ data::Degree::D,
data::AlterationSign::FLAT, data::Octave::LINE1 };
        data::Note expectedNote{ data::Degree::D,
data::AlterationSign::NONE, data::Octave::LINE1 };
        ++note;

        ASSERT_EQ(note, expectedNote);
    }

    {
        data::Note note{ data::Degree::E,
data::AlterationSign::SHARP, data::Octave::LINE1 };
        data::Note expectedNote{ data::Degree::F,
data::AlterationSign::NONE, data::Octave::LINE1 };
        ++note;

        ASSERT_EQ(note, expectedNote);
    }
}

/**
 * Описание: декрементация структуры Note
 * Когда: вызвана инкрементация оператор--
 * Тогда: нота декрементирована корректно
 */
TEST(NoteTest, Decrementation)
{
    {
        data::Note note{ data::Degree::C,
data::AlterationSign::NONE, data::Octave::LINE1 };
        data::Note expectedNote{ data::Degree::B,
data::AlterationSign::NONE, data::Octave::SMALL };
        --note;

```

```

        ASSERT_EQ(note, expectedNote);
    }

    {
        data::Note note{ data::Degree::D,
data::AlterationSign::FLAT, data::Octave::LINE1 };
        data::Note expectedNote{ data::Degree::C,
data::AlterationSign::NONE, data::Octave::LINE1 };
        --note;

        ASSERT_EQ(note, expectedNote);
    }

    {
        data::Note note{ data::Degree::E,
data::AlterationSign::SHARP, data::Octave::LINE1 };
        data::Note expectedNote{ data::Degree::E,
data::AlterationSign::NONE, data::Octave::LINE1 };
        --note;

        ASSERT_EQ(note, expectedNote);
    }
}

/**
 * Описание: сравнение структур Note
 * Когда: вызваны операторы сравнения operator> и operator<
 * Тогда: ноты сравниваются правильно
 */
TEST(NoteTest, Comparement)
{
    {
        data::Note note1{ data::Degree::E,
data::AlterationSign::SHARP, data::Octave::LINE1 };
        data::Note note2{ data::Degree::E,
data::AlterationSign::NONE, data::Octave::LINE1 };

        ASSERT_TRUE(note1 > note2);
        ASSERT_FALSE(note1 < note2);
    }

    {
        data::Note note1{ data::Degree::D,
data::AlterationSign::NONE, data::Octave::LINE1 };
        data::Note note2{ data::Degree::E,
data::AlterationSign::NONE, data::Octave::LINE1 };

        ASSERT_TRUE(note1 < note2);
        ASSERT_FALSE(note1 > note2);
    }

    {
        data::Note note1{ data::Degree::C,
data::AlterationSign::SHARP, data::Octave::LINE2 };
        data::Note note2{ data::Degree::C,
data::AlterationSign::NONE, data::Octave::LINE1 };

        ASSERT_TRUE(note1 > note2);
        ASSERT_FALSE(note1 < note2);
    }

    {
        data::Note note1{ data::Degree::F,
data::AlterationSign::SHARP, data::Octave::LINE1 };
        data::Note note2{ data::Degree::A,
data::AlterationSign::FLAT, data::Octave::SMALL };

        ASSERT_TRUE(note1 > note2);
        ASSERT_FALSE(note1 < note2);
    }
}

```

## Cmake:

```

cmake_minimum_required(VERSION 3.5)

if (WIN32)
    project(ElegiaApp VERSION 0.1 LANGUAGES CXX)
elseif(UNIX)
    project(ElegiaApp)
endif()

set(CMAKE_INCLUDE_CURRENT_DIR ON)
set(CMAKE_AUTOUIC ON)
set(CMAKE_AUTOMOC ON)
set(CMAKE_AUTORCC ON)
set(CMAKE_PREFIX_PATH $ENV{QTDIR})

find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Core)
find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Widgets)
find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Multimedia)
find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS MultimediaWidgets)
find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Gui)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS Core)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS Widgets)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS Multimedia)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS MultimediaWidgets)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS Gui)

set(QT_USE_QTMULTIMEDIA TRUE)
set(QT_USE_QTMULTIMEDIAWIDGETS TRUE)

add_subdirectory(backend)
set(CMAKE_CXX_FLAGS "-L ${BACKEND_FOLDER_PATH}/build/conanlibs")
set(CMAKE_CXX_STANDARD 17)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

message("Current Qt version is: ${QT_VERSION_MAJOR}.${QT_VERSION_MINOR}")

set(UI_DIR
    ui/mainwindow.ui
)

```

```

set(INCLUDE_H_DIR
    include/MainWindow.h
    include/NotesIdentifierController.h
    include/AudioEngine.h
    include/TonalityBuilderController.h
    include/TransposerController.h
)

set(SOURCES_CPP_DIR
    src/main.cpp
    src/MainWindow.cpp
    src/NotesIdentifierController.cpp
    src/AudioEngine.cpp
    src/TonalityBuilderController.cpp
    src/TransposerController.cpp
)

set(RESOURCES_DIR
    resources/Audios.qrc
    resources/Images.qrc
    resources/Styles.qrc
)

set(CMAKE_AUTOUIC_SEARCH_PATHS "ui")

if (WIN32)
    qt_add_executable(${PROJECT_NAME}
        WIN32
        MANUAL_FINALIZATION
        ${INCLUDE_H_DIR}
        ${SOURCES_CPP_DIR}
        ${UI_DIR}
        ${RESOURCES_DIR}
    )
    qt_finalize_executable(${PROJECT_NAME})
elseif(UNIX)
    qt_add_executable(${PROJECT_NAME}
        MANUAL_FINALIZATION
        ${INCLUDE_DIR}
        ${SOURCES_DIR}

```

```

    ${UI_DIR}

    ${RESOURCES_DIR}

)

qt_finalize_executable(${PROJECT_NAME})

endif()

target_compile_features(${PROJECT_NAME} PUBLIC cxx_std_17)

include(${BACKEND_FOLDER_PATH}/build/conanbuildinfo.cmake)

conan_basic_setup()

target_link_libraries(${PROJECT_NAME}

    PUBLIC

        elegia::utils

        elegia::data

        elegia::elegia

    PRIVATE

        Qt${QT_VERSION_MAJOR}::Widgets

        Qt${QT_VERSION_MAJOR}::Core

        Qt${QT_VERSION_MAJOR}::Multimedia

        Qt${QT_VERSION_MAJOR}::Gui

)

# Precompiled header

target_precompile_headers(${PROJECT_NAME} PRIVATE src/pch.h)

```