

Institiúid Teicneolaíochta Cheatharlach



At the Heart of South Leinster

Design Manual

Institute of Technology, Carlow

B.Sc.(Honours) in Software Development

CW238

Project Title

Musical Tablature Editor

Document Date

11/01/13

Student Name and ID

Robert Connolly C00123951

Supervisor

Paul Barry

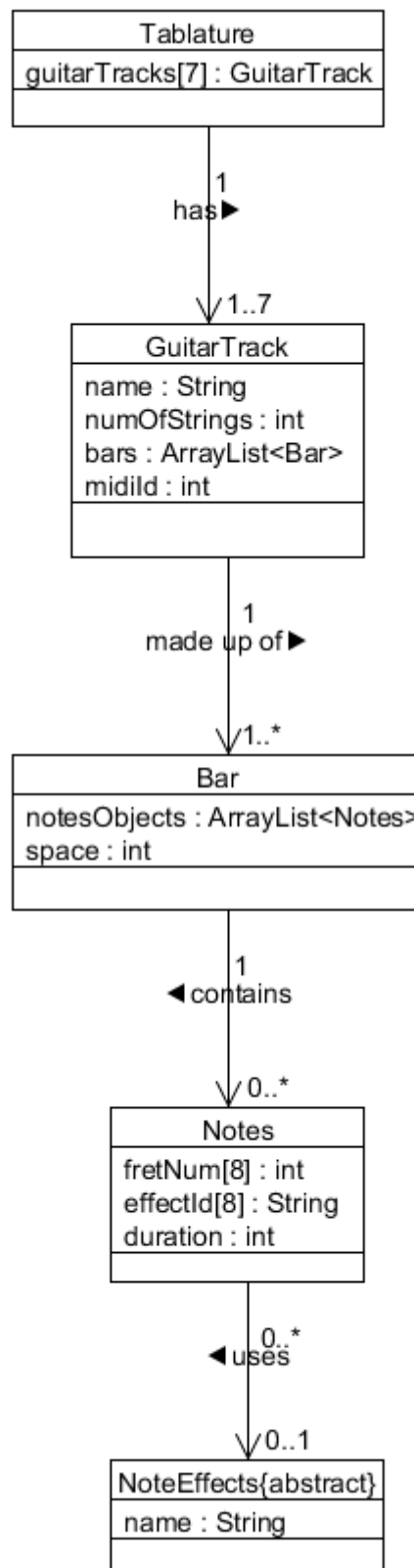
Table of Contents

Introduction	3
Class Diagram	4
Classes	5
1. Notes	5
2. Bar	5
3. GuitarTrack.....	6
4. Tablature	6
5. NoteEffects{Abstract}.....	6
Core Features Design.....	7
1. Electric Guitar Tablature.....	7
GUI's fretboard	7
GUI's tablature	8
2. Note Effects and Duration	10
2.1. Note Effects.....	10
2.2. Note Duration	10
3. MIDI Playback	12
Technologies used	13
Java Swing	13
Java Sound.....	13

Introduction

This document describes the design of the three core features of my project. It contains a class diagram that shows the classes that I will be using, and a description of each of the classes and their attributes. It also describes the technologies I will be using to develop the project.

Class Diagram



Classes

1. Notes

This class is used to contain the notes a user inputs to the tablature at any one time.

Attributes:

- **fretNum[8]:** Contains the fret number the notes are played on. Each index of the array corresponds to a string on the guitar. The fret number of a string will be put in the appropriate index for a string. A guitar can have up to eight strings.
Data type: int
- **effectId[8]:** Contains the id of the effect applied to each note.
Data type: String
- **duration:** This contains the length of how long the Notes objects are played. They are measured in up to sixty-four units. This is used because this is the shortest duration a Notes object can be played, therefore, allowing for a unit of measure for Notes objects with a duration longer than this. Here is the list of all the Notes objects duration using the sixty-four units as a unit of measure:
 1. Whole note = 64 units
 2. Half note= 32 units
 3. Quarter note= 16 units
 4. Eighth note= 8 units
 5. Sixteenth note= 4 units
 6. Thirty-Second note= 2 units
 7. Sixty-Fourth note= 1 unit

These measurements will be used to determine how long a Notes object is played in MIDI. It's also used to calculate how much space is left in a bar for more Notes objects.

The default value is a Quarter note= 16 units.

Data type: int

2. Bar

Attributes:

- **notesObjects:** Contains all the Notes objects in the bar in the consecutive sequence they appear in the bar. It is an arraylist in order to accommodate the changing amount of Notes objects that can be in a bar. The amount can be from one to sixty-four Notes objects depending on the duration of each of the Notes objects.
Data type: ArrayList<Notes>
- **space:** This contains how much space is left in a bar to accommodate more Notes objects. It is calculated by adding up the duration of all the Notes objects in a Bar

object and subtracting them from sixty-four. The space is measured in how many units of sixty-four units are left in the bar. **Data type:** int

3. GuitarTrack

Attributes:

- **name:** Contains the name of the instrument being tablatured.
Data type: String
- **numOf Strings:** Contains the number of strings of the instrument in the track. Used for building the GUI's number of strings.
Data type: int
- **bars:** Contains all the Bar objects in the track in consecutive order.
Data type: ArrayList<Bar>
- **midId:** Contains the ID of the MIDI soundbank applied to the track.
Data type: int

4. Tablature

Attributes:

- **guitarTracks[7]:** An array that contains each GuitarTrack object in the tablature. A total of up to seven guitar tracks can be used in the tablature.
Data type: GuitarTrack

5. NoteEffects{Abstract}

Attributes:

- **name:** Contains the name of an effect.
Data type: String

Methods:

- drawEffect()

This is a superclass for each note effect. Different note effects will inherit from this class.

Core Features Design

1. Electric Guitar Tablature

This design describes how a user adds, updates and deletes notes on the GUI's fretboard and tablature.

GUI's fretboard

This design describes how to add, change and delete notes on the GUI's fretboard.

- **Add a Note:**

1. The user uses the mouse cursor to add a note to the fretboard.
2. They place the cursor over a fret of a string on the fretboard, and clicks it with the mouse.
3. A dot is displayed over the string on the fret they chose on the fretboard.
4. The user can repeat 1-3 as many times as they like.
 - a. If they select a fret on a string that already has a fret selected, see *Change a Note*.

- **Update a Note:**

The user wants pick a different note to what they've previously selected on a string.

1. The user uses the mouse cursor to add a different note to a string.
2. They place the cursor over a different fret on the string on the fretboard, and clicks it with the mouse.
3. A dot is displayed over the new fret on the string on the fretboard.
4. The dot for the previous note is deleted from the fretboard.

- **Delete a Note:**

1. The user uses the mouse cursor to delete a note from the fretboard.
2. They place the cursor over the dot of the selected fret on the fretboard, and click the left mouse button.
3. The dot is removed from the fretboard.

GUI's tablature

This design describes how to add, update and delete notes on the GUI's tablature.

- **Add Notes:**

This design starts after the user has selected what notes they want on the GUI's fretboard.

1. The user clicks the 'Add Notes' button with the mouse.
2. A Notes object is created to store the notes.
3. The fret number and string number for each note is read from the GUI's fretboard.
4. The fret number is stored in an array, in the Notes object, in the appropriate index corresponding to the string number it is on.
5. The space left in the current Bar object is checked.
6. If there is space, but it is less than the duration of the Notes object, change the duration of the Notes object to the next lowest duration that will fit in the Bar object.
7. If there is no space left in the Bar object, create a new Bar object.
8. Store the Notes object in the next free index in a list in the current Bar object. This keeps track of what Notes objects are played in the Bar object in sequential order.
9. Update the space left in the Bar object.
10. The next free slot in the Bar object on the GUI's tablature is created to display the notes.
11. Display the notes on the strings of the GUI's tablature using the array in the Notes object. The notes are to be displayed by their fret number on the appropriate strings vertically.
12. Check the space left in the Bar object. Using the duration of the Notes object, add and display as many free slots in the GUI's tablature that can fit in the space left in the Bar object, directly after the Notes object. This shows the user how much space is left in the Bar object for notes of the duration that they added.

- **Update Notes:**

This is the design for when a user wants to update notes on the GUI's tablature.

1. The user clicks on a note in the GUI's tablature.
2. The note and any other notes played at the same time are displayed on the GUI's fretboard, from the array in the Notes object that contains the note.
3. The user changes a note, see *Update a note* in *GUI's fretboard* design. The user deletes a note, see *Delete a note* in *GUI's fretboard* design.
4. When finished, the user presses the 'Add Notes' button.
5. The original notes in the array in the Notes object, are updated to what's on the GUI's fretboard. See steps 3 and 4 in *Add Notes*.

6. The updated notes are displayed in place of the original notes on the GUI's tablature.

- **Delete Notes:**

1. The user clicks on a note on the GUI's tablature.
2. The user clicks the 'Delete Notes' button.
3. The Notes object, containing the note and any other notes that were played with it, is destroyed and deleted from the list of the Bar object that contains it.
4. Re-display the Bar object on the GUI's tablature using the updated list in the Bar object.
5. Check the space left in the Bar object. Using the duration of the last Notes object in the Bar objects list, add and display as many free slots in the GUI's tablature that can fit in the space left in the Bar object, directly after the last Notes object. This shows the user how much space is left in the Bar object for notes of the duration of the last Notes object in the bar.

2. Note Effects and Duration

This design describes how to add, update and delete note effects and notes durations on the GUI's tablature.

2.1. Note Effects

This design describes how to add, update and delete note effects on the GUI's tablature.

- **Add Effects:**

1. The user clicks on a note in the GUI's tablature.
2. Then clicks one of the note effect buttons.
3. The name of the effect is stored in the effects array of the Notes object that contains the note, in the appropriate index for the note's string number.
4. The effect is drawn beside the note on the GUI's tablature using the note effect's draw method.

- **Update Effects:**

1. The user clicks on a note in the GUI's tablature that already has a note effect.
2. Then clicks on a different note effect button.
3. The new note effect's name is stored in the effects array of the Notes object that contains the note, in place of the original.
4. The old effect is removed from the note on GUI's tablature.
5. The new effect is drawn beside the note on the GUI's tablature using the new note effect's draw method.

- **Delete Effects:**

1. The user clicks on a note on the GUI tablature that has a note effect they want to delete.
2. Then clicks the same note effect button for the effect already applied to the note.
3. The note effect's name is removed for the appropriate note, from the effects array of the Notes object that contains the note.
4. The note effect is deleted from the note in the GUI's tablature.

2.2. Note Duration

This design describes how to update a note's duration in the GUI's tablature. Any notes played at the same time will have the same duration. A note always starts with a default duration of a quarter note when added to the tablature. A note's duration can never be deleted.

Update Note Duration:

1. The user clicks on a note in the GUI's tablature.
2. Then the user clicks one of the duration buttons that has a different duration than what is already applied to the Notes object that contains the note.

3. The space that is left in the Bar object that contains the note is checked.
4. If the new duration is less than or equal to what space is left in the Bar object, the duration of the Notes object that contains the note is changed to the new duration value. Otherwise, message the user the new duration is too long for the bar.
5. If the duration has changed, check the space left in the Bar object. Using the changed duration of the Notes object, add and display as many free slots in the GUI's tablature that can fit in the space left in the Bar object, directly after the Notes object. This shows the user how much space is left in the Bar object for notes of the duration that they added.

3. MIDI Playback

This design describes how the user plays back their tablature using MIDI.

Playback from beginning of tablature:

1. The user clicks the “playback from start” button. Playback will start at the very beginning of the tablature.
2. Start with the first Bar object in the tablature.
3. Start with the first Notes object in the bar.
4. Retrieve the string number and fret number of each note in the Notes object.
5. Using the note value of the bottom string of the GUI’s fretboard as the lowest note, calculate the note value of each note in the Notes object based on its string number and fret number in relation to the lowest note.
6. Retrieve the duration of the Notes object. Convert it to microseconds. This will be used for how long each note in the Notes object are played in the playback.
7. Retrieve the index of where the Notes object is in the list containing it in the current Bar object. This index, plus 1, will be on which beat in the tablature each note in the current Notes object is played.
8. Pass the note value of each note, the duration of each note and on what beat it is to be played on to the MIDI device and play the notes.
9. Go to the next Notes object in the current Bar object.
10. Repeat steps 4-9 for all the Notes objects in the current bar.
11. Go to the next Bar object in the tablature and repeat steps 4-10 until there are no more Bar objects left in the tablature.

Playback from cursor:

1. The user clicks any note in the tablature with the mouse cursor.
2. The user clicks the “playback from cursor” button. Playback will start from the Notes object that contains the note the user selected with the mouse cursor.
3. Repeat steps 4-11 in *Playback from beginning of tablature*.

Technologies used

Java Swing

This is the GUI library I will be using to implement my GUI. I am using it due to its portability. It also has large documentation resources that will make it a great tool to learn, develop and implement my GUI with.

Java Sound

This is the standard sound library that comes with the Java SDK. It contains all the functionalities I need for playing back the tablature in my project through MIDI. It too has large documentation resources for it, making it a great tool to learn, develop and implement the MIDI part of my project with.