COMP6000 Project Logbook

Hannah Rhodes – hjr22

Table of Contents

Task 1: Getting the jMusic library set up………………………………………………………………………………3

Task 2: Reading through and following some tutorials on the jMusic web page…………………..3

Task 3: 1st Meeting with Anna……………………..……………………………………………………………………….3

Task 4: Use the jMusic library to play a simple song and save it to a midi file……………………… 3

Task 5: Add a basic harmony to ‘Happy Birthday’…………………………………………………………………4

Task 6: Create a piece with notes from a specific scale and then add a harmony…………………4

Task 7: 2nd Meeting with Anna……………………………………………………………………………………………..5

Task 8: Make a 2D array in my ‘Scale’ project rather than multiple 1D arrays……………………….5

Task 9: 3rd Meeting with Anna………………………………………………………………………………………………5

Task 10: Fix the loop so that phrases[0] and phrases[1] can be played as a harmony……………6

Task 11: Fix the loop structure to allow the correct number of notes to be added……………….6

Task 12: Find out why the loop occasionally adds an extra note to the phrase and fix it………7

Task 13: 4th Meeting with Anna…………………………………………………………………………………………….7

Task 14: Create chords using my existing loop………………………………………………………………………7

Task 15: 5th Meeting with Anna…………………………………………………………………………………………….8

Task 16: Use CPhrase to create different types of chords……………………………………………………..9

Task 17: 6th Meeting with Anna…………………………………………………………………………………………….9

Date: 4/10/2022

Tasks:

* Task 1: Getting the jMusic library set up
* Task 2: Reading through and following some tutorials on the jMusic web page : [jMusic Tutorial Index (explodingart.com)](https://explodingart.com/jmusic/jmtutorial/t1.html)

Comments:

* Task 1: I had some issues originally setting up jMusic on IntelliJ (my chosen IDE) since I tried to add the library through downloading the files and trying to import them as a library. I didn’t realise until after trying the first way for a while that I could import the library through the Maven repository. The library worked after I tried this.
* Task 2: I read through all the tutorials on the jMusic web page, and I followed a basic tutorial called ‘Bing’ which plays a single midi note.

Next task:

* 1st Meeting with Anna

Date: 4/10/2022

Tasks:

* Task 3: 1st Meeting with Anna

Comments:

* Task 3: At our meeting we discussed the troubles I had setting up the jMusic library and the ‘Bing’ task I completed following successful setup of the library. We also discussed some possible wider ideas about the potential things I could base my project on. I said that I wanted to do something with harmonies, but I wasn’t completely sure about specific ideas yet. We said that we would talk in our next meeting about shaping the project down to a specific idea. I also said I wanted to play around more with the jMusic library to get more familiar with it.

Next task:

* The next task I have set is to do a slightly more complex program that will play a simple song.

­­­­­­­­­­­­­­­­­­­­­­­­

Date: 5/10/2022

Tasks:

* Task 4: Use the jMusic library to play a simple song and save it to a midi file.

Comments:

* Task 4: Using the jMusic library and the knowledge I acquired from doing the ‘Bing’ project I was able to use an array of notes and durations of those notes to play the song ‘Happy Birthday’ with just single notes. I then was able to save it as a midi file. I had downloaded QuickTime (a media playing software) as I thought I would need it to hear the midi file being played but I was able to hear it through Windows Media Player.

Originally the song was playing extremely slowly so I needed to find a way to increase the tempo, which I did manage to do.

Next task:

* For my next task I want to use the jMusic library to make an even more complex song, add a basic harmony since as I have only managed so far play single notes in the song. I might try and extend my ‘Happy Birthday’ song to include a harmony.

Date: 8/10/2022

Task:

* Task 5: Add a basic harmony to ‘Happy Birthday’

Comments:

* Task 5: I made another array for one harmony and firstly tried to add it to the parameter that contained the melody array, but this ended up playing the harmony notes straight after the melody notes. Instead, I created another ‘phrase’ with the harmony notes that contained a copy of the melody phrase, so the notes would play at the same time. This worked so I then added a third harmony to the previous 2 arrays, again creating the new phrase that contained a copy of the last phrase with the melody and first harmony already in it.

Next task:

* The next task I might try is to create a piece of music containing only notes from a specific scale and then add a harmony in the same scale to it.

Date: 9/10/2022

Task:

* Task 6: Create a piece with notes from a specific scale and then add a harmony

Comments:

* Task 6: I started this task by creating a new part, a new phrase, and a new score. I created some extra variables, one to hold a random number, and used the predefined field for a pentatonic scale. I created a loop which would generate a note from the scale randomly using the random number but within a range avoiding very high or very low notes and add the note to a phrase, repeating this until it reached 20 (the number I used for the number of notes in the piece).

I tried adding another phrase using the same loop, but this did not create the harmony, it just doubled the length of the piece. I had to use a separate for loop for the second phrase and create a copy of phrase1 to add to phrase 2 so the harmony notes would play at the same time.

Next task:

* 2nd Meeting with Anna

Date: 11/10/2022

Task:

* Task 7: 2nd Meeting with Anna

Comments:

* Task 7: In our meeting I showed Anna what I had done so far, and we discussed where we could take what I had done already and incorporate this into my idea. We talked about taking my ‘scale’ project with two 1D arrays and making a 2D array with the phrases as a starting point for further additions to the project. For example, being able to add constraints to the loops to represent different musical grammars. By doing this I would be able to create harmonies for different music styles. We talked about how I could possibly make this into an app that has buttons for different music styles e.g., blues, classical etc. and when pressed would generate a harmony in that style. For the interface of the app, we talked about using Android Studio or Java web app but we decided to keep this idea up in the air for now so I can work on the back end.

Next task:

* Make a 2D array in my ‘Scale’ project rather than multiple 1D arrays

Date: 12/10/2022 – 17/10/2022

Task:

* Task 8: Make a 2D array in my ‘Scale’ project rather than multiple 1D arrays

Comments:

* Task 8: I have been trying to create a 2D array for the ‘Scale’ project called ‘Scale2’, however I have been having a lot of trouble with type errors. I tried using a 2D int array (int[2][20]) where the first array is the number of phrases, and the second array was an array of 20 notes but when trying to use the output of the array in a Phrase I ran into type errors where it could not convert an int to a phrase. I decided to purchase the book “Making Music with Java: An Introduction to Computer Music, Java Programming and the JMusic Library” by Andrew Brown to see if this will help me with using the JMusic library

Next task:

* 3rd Meeting with Anna

Date: 18/10/2022

Task:

* Task 9: 3rd Meeting with Anna

Comments:

* Task 9: In our meeting we looked at the code in the ‘Scale2’ where I was unsuccessful at making a 2D array. Initially I tried to get a 2D array working with a Phrase object ‘Phrase[2][20]’ where [2] was the number of phrases and [20] was each note in the phrase, however this did not work. I recognised that I didn’t need an array to hold the number pf phrases and another array to hold the number of notes in the phrase since the ‘Phrase’ object already has the capacity to store the notes within it, therefore I only needed a single array, and accessed the notes through a single array of phrases called ‘phrases[2]’. Phrases[0] and phrases[1] are the indexes that each store several notes, in this case it was 20 notes, and assigned the array of phrases to a Phrase object called phr1.

While I got as far as being able to play phrases[0], I didn’t get around to playing phrases[1] because of some problems with the phr1 in the loop, I had initially tried to create a new phrase at the end of the loop to assign the phrases[1] to when phrases[0] was full but instead this just overwrote phr1 and also phrases[1] so when the midi file was played it was only playing the notes in phrases[0].

Next task:

* The next task is to fix the loop so that phrases[0] and phrases[1] can be played together as a harmony

Date: 19/10/2022 – 24/10/2022

Task:

* Task 10: Fix the loop so that phrases[0] and phrases[1] can be played together as a harmony

Comments:

* Task 10: I struggled with getting the loop to create new note values for each index for phrases[]. Originally, the loop was adding notes to phrases[0] (In the case of testing I used 3 notes), however when phrases[0] was finished and incremented to phrases[1] the loop took the three notes from phrases[0] and added them to phrases[1] and then added 3 more notes to both phrases[1] and phrases[0]. I worked out this was because I was putting “phrases[noOfPhrases] = phr1” the end of the loop, so the loop was assigning both phrases[0] and phrases[1] to phr1 making them both the same.

I tried making a new phrase object at the end of the loop, but this just overwrote whatever was added to phrases[0] to 0 and then did the same for phrases[1].

I managed to eventually make a new phrase at the beginning of the loop which stopped the problem of both phrases having the same values however, this introduced a new problem in which the loop is not adding the correct amount of notes to each phrase and was instead using the length of the pentatonic scale array (5 notes) as the number of notes to add to the phrase, but also using the line of code that generates the random number to decide how many notes out of 5 it should add to each phrase. This is a problem with the structure of the loop which needs to be fixed by re-arranging it.

Next task:

* 4th Meeting with Anna (postponed)
* Fix the loop structure to allow the correct number of notes to be added to each phrase

Date: 30/10/2022

Task:

* Task 11: Fix the loop structure to allow the correct number of notes to be added

Comments:

* Task 11: I was able to re-arrange the loop to allow a pre-specified int of notes to be added to the phrase, rather than adding the same number of notes as the length of the scale. Whilst most of the time this works there are occasional times where it adds one more note than the pre-specified int so for example if the pre-specified int was 3, it will add 3 notes to phrase[0] but 4 notes to phrase[1]. I thought maybe this was to do with the loop having ‘notesInPhrase < noteNum’ rather than ‘notesInPhrase <= noteNum’ but when I changed it, I still encountered this problem.

Next task:

* Find out why the loop occasionally adds an extra note to the phrase and fix it

Date: 5/11/2022

Task:

* Task 12: Find out why the loop occasionally adds an extra note to the phrase

Comments:

• Task 12: After debugging my code I noticed that the occasional problem of having one to many notes added to the phrase happens when the loops finds a successful note near the start of pent[p] e.g. when pent[p] = 1, and another successful note before pent[p] has finished its loop e.g. when pent[p] = 5 (containing the number of notes in the scale, in the testing case this is 7 notes for the harmonic minor scale). In other words when a random number that is generated contains more than one successful note when calculating the 7 different moduli in the single loop. When this happens, it ends up adding an extra note to the phrase, but not every time as it is just down to chance that one random number will have two successful moduli.

I added a break after the last for loop in the nested loop to jump out of the loop once it reaches noteNum and doesn’t complete the rest of the iterations through pent[p] which fixed the problem and now the loop works as it should.

Next task:

* 4th Meeting with Anna

Date: 8/11/2022

Task:

* Task 13: 4th Meeting with Anna

Comments:

* Task 13: In our meeting we discussed how far I had gotten with the loop, and we discussed where next to take things. We talked about using the loop I had created to create chords which would make things easier in the long run when I get to dealing with different styles of music and their characteristics. By being able to create chords I will then be able to use those chords inside methods that deal with other chords that might not work together rather than struggling with complicated code which would make it harder to see the chords that I am working with. We discussed that by our next session I should try and create just a few chords using the loop I made, where the loop will generate notes and only add the correct ones to the array, just like having an array for correct notes from a scale, I can define the array for a chord for example the note F which has notes [F,A,C]. By defining the chord in the array declaration and then using the loop to generate the chord, it should only add notes F,A and C to the array.

Next task:

* Create chords using my existing loop

Date: 9/11/2022 – 14/11/2022

Task:

* Task 14: Create chords using my existing loop

Comments:

* Task 14: I have been having some trouble using my existing loop to create chords. I tried to make three phrases each with one note each. I first changed the int [] pent = SCALE to have a specific set of values instead of an empty array. I used an f note with notes F, A and C (I used the midi number values) which I called fNote [].

The first problem I had was the loop was randomly generating a number, the result gave a random mixture of notes, some duplicates and also missing notes so I removed the random number generation and added a for loop that starts at 0 and loops through all 127 midi notes, so that when the counter got to one of the correct notes, it would add it and then carry on from where it left off, up to the next note until it has added all the notes in order.

The next problem I had was that instead of adding one note to each phrase, it was adding all 3 notes to each of the 3 phrases in order, so phrases [0] had {F, A, C}, phrases [1] had {F, A, C} and so on. I realised that this was because the loop “for (int p = 0; p < fNote.length; p++)” was incrementing p three times before incrementing the phrase, so I removed this because it was unnecessary. By doing this it did fix the problem of having 3 notes in each of the 3 phrases, and now each phrase does only have one note however, now the notes all have a pitch of 0 because there is not a reference to fNote.

Next task:

* 5th Meeting with Anna
* Fix the problems with the chord loop

Date: 15/11/2022

Task:

* Task 15: 5th Meeting with Anna

Comments:

* Task 15: In our meeting we looked at the code I had done on chords, and we discussed a way that I can make a chord without using a loop. In the code I had already done I was trying to access variables in the array and compare them to an index which was why empty notes were being added to the phrases. We looked at the CPhrase class of JMusic and Anna helped explain what the code was doing in terms of chord theory which I was not understanding before. I think that using the CPhrase class will be very helpful for making chords as each type of chord (e.g., major, minor, 7ths etc) have their own specific note differences for each note. This way I can create methods to make major chords by taking the root note and adding root note + 4 and root note + 7 as this is the same for all major chords. Other chord types have a different set of differences. I can then make multiple methods to create different types of chords.

Next task:

* Use CPhrase to create different types of chords

Date: 16/11/2022 – 21/11/2022

Task:

* Task 16: Use CPhrase to create different types of chords

Comments:

* Task 16: I managed to implement a method using CPhrase for major chords. I used separate variables with the midi numbers for each note from A to G so I could use this as a parameter for the starting note. I then used the starting note, starting note +4 and starting note +7 to create a major chord. I ran into a problem with writing the chord to a midi file as it had said it had written it but when I tried to play the file there was no sound.

Next task:

* 6th Meeting with Anna

Date: 29/11/2022

Task:

* Task 17: 6th Meeting with Anna

Comments:

* In our meeting I showed Anna what I had done with the CPhrase for major chords, and that it was writing it to a file but not playing. We found an easier way to play the chords without having to save it to a file because this can be frustrating to keep opening and closing the media player each time I change the code. We implemented the ‘Play.midi(part)’ which plays the sound immediately without having to save it to a file to play it. I then created a new method the same as the ‘major’ method but with a slight difference in the pattern (i.e. using the starting note, starting note +3 and starting note +7) to make a minor chord. I realised that there would be a problem with the ‘Play’ part this once I start to implement other chord creating methods because there was a ‘Play’ section in each method but no way to use it in the main method, so I created a separate method for Play called playChord(part p) where in the main method this can be used to access any of the other chord creation methods. After doing this we used playChord to play the chords for ‘Happy Birthday’

Next task: