BSPro - A First Bachelor Semester Project in BiCS-land

Monday 20th December, 2021 - 14:05

Stefan Capitanescu
University of Luxembourg
Email: stefan.capitanescu.001@student.uni.lu

This report has been produced under the supervision of:
Nicolas Guelfi
University of Luxembourg
Email: nicolas.Guelfi@uni.lu

Abstract

This document is a template for the scientific and technical (S&T for short) report that is to be delivered by any BiCS student at the end of each Bachelor Semester Project (BSP). The Latex source files are available at: ***https://github.com/nicolasguelfi/lu.uni.course.bics.global***

This template is to be used using the Latex document preparation system or using any document preparation system. The whole document should be in between 6000 to 8000 words [1] *(excluding the annexes) and the proportions must be preserved. The other documents to be delivered (summaries, ...) should have their format adapted from this template.*

A tutor (or any person having contributed to the BSP work) is not a co-author per se for a student's work. It is possible to exploit a BSP report to produce a scientific and technical publication. In this case, the authors list has to be discussed and agreed with the concerned parties.

1. Plagiarism statement

This 350 words section without this first paragraph must be included in the submitted report and placed after the conclusion. This section is not counting in the total words quantity.

I declare that I am aware of the following facts:

- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:

1) Not putting quotation marks around a quote from another person's work
2) Pretending to paraphrase while in fact quoting
3) Citing incorrectly or incompletely
4) Failing to cite the source of a quoted or paraphrased work
5) Copying/reproducing sections of another person's work without acknowledging the source
6) Paraphrasing another person's work without acknowledging the source
7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
8) Using another person's unpublished work without attribution and permission ('stealing')
9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

1. i.e. approximately 12 to 16 pages double columns excluding the Plagiarism Statement

2. Introduction ($\pm$ 5% of total words)

This paper presents the bachelor semester project made by Motivated Student together with Motivated Tutor as his motivated tutor. It presents the scientific and technical dimensions of the work done. All the words written here have been newly created by the authors and if some sequence of words or any graphic information created by others are included then it is explicitly indicated the original reference to the work reused.

This report separates explicitly the scientific work from the technical one. In deed each BSP must cover those two dimensions with a constrained balance (cf. [BiCS(2021)]). Thus it is up to the Motivated Tutor and Motivated Student to ensure that the deliverables belonging to each dimension are clearly stated. As an example, a project whose title would be "PLAYTOUCH - A multi-user game for multi-touch devices" could define the following deliverables:

- Possible scientific [Armstrong and Green(2017)] deliverables:
  - What are concurrency models and how are they implemented?
  - How is measured ergonomics in human-computer interaction?
  - How to model the concurrency of a multi-touch devices?
  - Can PLAYTOUCH enter in a blocking state?
  - How to model the design of PLAYTOUCH?
- Possible Technical deliverables:
  - PLAYTOUCH Implementation
  - PLAYTOUCH Tests implementation
  - Hardware end system configuration for PLAYTOUCH

The length of the report should be from 6000 to 8000 words excluding images and annexes. The sections presenting the technical and scientific deliverables represent $\pm$ 80% of total words of the report.

3. Project description ($\pm$ 10% of total words)

3.1. Domains

3.1.1. Scientific . Provide a short description of the scientific domain(s) in which the project is being made.

3.1.2. Technical. Provide a short description of the technical domain(s) in which the project is being made.

3.2. Targeted Deliverables

3.2.1. Scientific deliverables. Provide a synthetic and abstract description of the scientific deliverables that have been produced. Each BSP must contain some work done according to the principles of the scientific method. It basically means that you should define at least one question related to the knowledge domain of your BSP and follow part of the scientific method process to answer this question. The description of the work done to answer this question is a scientific deliverable.

Other examples of question could be:

- Is Python an adequate language for concurrent programs?
- How can we measure the ergonomics of a graphical user interface?
- How can we ensure that a program will not fail?

An answer to such question should be the result of applying partly or totally the scientific method according to its standard definition which can be found in the literature.

As you can see in this template, the scientific deliverable is entirely separated from the technical deliverable. It the default case it addresses a question closely related to the technical deliverable.

3.2.2. Technical deliverables. For this bachelor semester project, the target to reach is to create a software that could export an audio file of none, one or multiple songs which are defined by different criteria.

The program was planned to be written in python, a world-known programming language open-source which is widely used for its versatility and various libraries and modules. For this project, the musical module "pydub" had to be used to import and export songs with only the path of the audio file searched. Even if this module has more interesting features they normally will not be used for the code. A graphical user interface was expected to be done for the BSP with Tkinter, an python blinding easy to use and with a great documentation all over the internet. Provide a synthetic and abstract description of the technical deliverables that were targeted to be produced. A technical deliverable in this report is the description of a product build by the student using software or hardware technologies.

4. Pre-requisites ($[5\%..10\%]$ of total words)

Describe in these sections the main scientific and technical knowledge that is required to be known by you before starting the project. Do not describe in details this knowledge but only abstractly. All the content of this section shall not be used, even partially, in the deliverable sections. It is important not to include in this section all the knowledge you have been obliged to acquire in order to produce the deliverable. It should only state the knowledge the student possessed before starting the project and that was mandatory to possess to be capable to produce the deliverables. It explicitly defines the technical and scientific pre-condition for the project. It is also useful to avoid project failures due to over or under complex subjects.

4.1. Scientific pre-requisites

4.2. Technical pre-requisites

5. A Scientific Deliverable 1

For each scientific deliverable targeted in section 3
2 provide a full section with all the subsections described

below.

5.1. Requirements (± 15% of section's words)

This scientific deliverable will cover different questions but all related to the main question which is ”How can we evaluate the similarities between two instrumentals tracks”. All the work in the production section will divided into parts of different questions we will answer to and they are all related one to each other to have an overview of the main scientific question.

5.2. Design (± 30% of section's words)

This design section exist for the purpose of explaining the decisions made about the selection of topics in the production part and the decision made for the sources' section. The following subsection are there also to precise which main document are used in each production section. For each subsection of the scientific production, the design part explains the reason why the subsections are important, what are their relations between them, what are the main used documents in each part, why our decision steered for one document and not for another and finally how these documents can give answers to the questions asked in the technical production.

5.3. Topic selection

For the musical term subsection, it is a choice that could not be skipped since the technical deliverable implies to understand the concept of ”BPM” and scale. This section is also important to understand some notions for the last production part. This part is difficult to be summarized in few lines mainly because it implies for the reader to have a musical knowledge which is necessary to understand the multiple definitions. Anyways, the part cover only necessary scientific notions that need to be understood without knowing anything about music theory.

It was not necessary to search and write about digital sounds and to make the comparison with the analog but since this project is relying on a evaluation of similarities, it assumes the use of encoded audio files and not some analog sounds. The difficult part of those two sections is to be concise and to be short because the main part and the most complex is the next and final one : the music recognition.

The music recognition part is the most important section of the production part because it is the section that answers to the main question of this deliverable. This field of research is constantly proposing new approaches to recognize music tracks and so this subsection contains information about three of those recognition methods.

5.4. Source selection

5.5. Production (± 40% of section's words)

Musical Terms. As mentioned previously, this production section partly defines some of the most used musical terms. Those definitions are useful and necessary to fully understand this report part. Musical notes are notation which gives information to a musician or any person that can understand this notation. The notation gives the time signature, the”BPM” (beats per minute), the position in time of a sound, its duration, its reference note, its dynamic, its panning, its pitch variation and some other aspects that are less often used. It's possible from those information to interpret them and extrapolate to know for example the scale of an instrumental.

There exist several way of writing music but the most popular in Europe and in North America remains : Western musical notation, also called music sheet, and the most recent Midi format (Musical Instrument Digital Interface) which implement more features and is the most used actually in non classical song to create sounds in a Digital Audio Workstation.

The position in time of a note is literally from when the note will be executed in a song. That notion introduces also the length or duration of that note, in simple words it is how long the note will be sounding. Even if some sounds does not have a very long sustain, the most important is the way it is played. For example, an harpsichord cannot have a very long sustain but the musician still can keep the finger on the touch. In Western Europe, since centuries the note reference are in the twelve equal temperament system, that means that from one note to its octave, which is a frequency rapport of two over one, there exist twelve frequencies equally separated. In anglophone regions, each note is assigned to a letter from ”A” to ”G”, with sometimes accidents which represent one equal or more temperament upper or lower.

But now let's come with the two of the most important musical notion in songs that are required in the production part : ”scale” and ”BPM”. Firstly, a scale is a set of music notes that are used in a music song. In the modern music, the most often used are the diatonic scale (scale that uses seven notes) but in other culture or more rarely musician uses other type of scale, for example the scale that is generally used for blues melodies is the pentatonic minor scale, a scale with only five notes. Scales can have any first note of the scale (called the root note) and have to follow after the relation of intervals between them.

Secondly, the ”BPM” is the acronym of beats per minute which represent the number of times the reference notes is executed in one minute. This is also an indicator to know if a song is relatively slow or relatively fast. Those two notions are very useful for disc-jockey (more commonly named ”DJ”) to know which songs to use when it comes to combine them or create transitions from one to the other.

Digital Sounds. As we know, the actual most listened music in the world are almost always going into an computer. That implies that the music nowadays is almost always digital. A sound is a vibration in the air and can be represented by a waveform.

Differences between Analog and Digital. An analogue sound has a natural and continuous waveform that created pressure and depression in the air and which has not been edited, a digital sound can also be represented as a waveform but this time it has been encoded with a certain limit of sample rate for every second and, depending on the format, the waveform can be compressed where analogue sound was totally continuous with no alteration. We can after distinguish two kind of digital audio file : compressed and lossless.

These two kinds of file have different uses, the compressed audio file was created, like a majority of other compressed file, to take less space in the storage, but the quality of the file is inferior to the lossless audio file. Since the creation of digital audio format, engineers analyses sound with the apparition of new type of digital measurements.

Music recognition. Since the starting of the millennium, engineers and scientists have been working for the past two decades on the music recognition. Constantly evolving, the different project became more and more efficient and quicker. The most known software for music recognition, Shazam, develloped by Shazam Entertainment Limited, is using a method to recognize songs called the ”fingerprint algorithm” with the help of considerable databases relying on.

This will be the base of our research to answer the question previously quoted, ”How can we evaluate the similarities between two instrumentals tracks” ? But in a first place, let's answer the question : ”How Shazam and other recognition software work” ?

In a way to analyse sound and get results, the software needs an input. Fortunately, nearly all the computers and mobile devices have a microphone input that can be used. The method of recognition a song by only a degraded sample is called the ”audio fingerprint” and it consists grossly of analysing from a point in time, which can graphically represent the frequency of a musical note or from a rhythm in time. Those points are replaced on a time-frequency graphic that can be visualized. (insert picture from main doc). Those points are replaced and analysed in a way to see if it correspond to a song or if it has a lot of same points. From one random point the algorithm analyses next musical notes that are detected by the microphone and which are in a similar range of frequencies. This methods is called the ”pattern matching” method. This is possible due to enormous databases which are created and optimized for those platforms with information needed for that process.

The formula of the audio fingerprint method rely on three detected musical notes n=(p,t) their pitches ”p” and their time ”t”. Consider n_1=(p_1,t_1), n_2=(p_2,t_2) and n_3=(p_3,t_3) three note detected by the input microphone and recognised by the device as a note. With $t_1 < t_2 < t_3$, the formula uses relations between two points in times and pitch relation. We define $dt_{1,2}$ = $t_2 - t_1$ the difference between the time of the second note and the time of the first note and $dt_{2,3}$ = $t_3 - t_2$ the difference between the time of the third note and the time of the second note. Additionally, we compute the difference between pitches the same way as the time measurement : $dp_{1,2}$ = $p_2 - p_1$ and $dp_{2,3}$ = $p_3 - p_2$. The audio fingerprint is defined as a list with three values :

[$dp_{1,2}$, $dp_{2,3}$,$dt_{1,2}$ dt$_{2,3}$]//

This analysed list and other captured audio fingerprint match with a similar or identical sheet music document from the database of those music recognition apps.//

There exist more recently a new approach that get a lot of interest in, it is the deep learning. The deep learning approach is a method that consists to create an algorithm that understand information expected from a query and to learn automatically from. The major disadvantage of this technique is the amount of data needed for the learning process. The way the method works here needs an audio sample excerpt from a song and the audio sheet of the song.//

We will not cover the

5.6. Assessment (± 15% of section's words)

Provide any objective elements to assess that your deliverable do or do not satisfy the requirements described above.

6. A Technical Deliverable 1

For each technical deliverable targeted in section 3
2 provide a full section with all the subsections described below. The cumulative volume of all deliverable sections represents 75% of the paper's volume in words. Volumes below are indicated relative the the section.

6.1. Requirements (± 15% of section's words)

The program produced for this bachelor semester project, is a python program running on the console that find songs in a folder dedicated and export some of them in one audio file with some defined user criteria selection.

FUNCTIONAL :

Import songs. The program get an sting input from the user, in our case it is in a plain text file, that contains the path to access an audio file. The result of that function is to import and open an audio file but without playing it.

Criteria selector. The program let the user to activate or not a criteria and to give it a value. The input expected from the user are, firstly, a ”Y” string to activate the mentioned criteria and a ”N” string to continue with other criteria, and a string for the criteria value except for the BPM where it's requested an integer. The result of those criteria will delete the not matching songs that were imported.

Concatenate songs. The concatenate songs function take as an input the remaining/selected opened songs from the import songs function and concatenated them one after the other. The end result is a variable that contains all the selected opened songs concatenated.

Export result. The export result function take as an input the variable containing all the concatenated selected songs and export them in one audio file.

NON-FUNCTIONAL :

Easy of use. The user has just to enter the name of the future exported file, ”Yes” or ”No” for selecting a criteria and to enter the values wanted. Every of those explained steps are clear and simple of comprehension.

Efficiency. The program has a while loop that does not allow to continue the program if at least one of the criteria was not selected. That prevents from exporting all the songs, which would not have any sense and to make the program useful.

6.2. Design (± 30% of section's words)

Before starting to code the actual program, a mock up was made using Miro, a free website and application whose main functionality is to let the user create a mock up with tools integrated. This platform is also used for meetings, brainstorming with other people and to manage and plan for a work team. Here is the realised mock up for this project :

(insert the mockup)

This platform was advised by the tutor of this project for the intuitive interface, the variety of tools proposed, and the convincing result a user can achieve without using it before. The realised mock up used only a few functionalities that were proposed because it was not an important part of this project and it has been used to show only how the program would work and ,if achieved, a design for the software produced. After that, this mock up is used as a template to remind the main function that the program offers and how it operates.

The actual program runs in python, an open source programming language that offers a diversity of very useful libraries when it comes to manipulates different type of files or creating a program that is not only based on the basic functionalities of python. For manipulating audio files there are several libraries that can be used, but for the purpose of this Bachelor Semester Project the simplicity and capability of the Pydub library corresponded to the program needs. Published firstly in 2011, Pydub created by James Robert has been upgraded and is still upgraded nowadays, for the purpose of this project the functionalities of importing, exporting and concatenating songs. Using this library has multiples advantages. Firstly, it is one of the most used libraries for audio file manipulations, it has a decent amount of documentation all over the internet and the commends are really intuitive and does corporate well with the python programming language.

To insert the data about the audio files containing tracks, we will use a simple text file. with on each line having the different information about the songs separated by a comma. This file is after interpreted when the user launches the program. This technique is a very good idea, instead having every description of every songs in the real python program and make it less readable, to have an external text file has several advantages and the open built-in function from python is clear and is an old functionality. Even thought the actual code is not complex and does not have a great length, it remains better to keep data in another file to not disturb the main program.

The program uses class to define criteria the project will rely on classes.

6.3. Production (± 40% of section's words)

cf. section **??** applied to the technical deliverable

6.3.1. Import songs. Firstly, before starting to use some specific function, the program has got a class where we call the python implemented function ”init”. We define the criteria as object to search them after in a while loop.

```
class Song:
    def __init__(self, bpm, scale,
    path, genre, composer, language,
    artist, songName):
        self.bpm = bpm
        self.scale = scale
        self.path = path
        self.genre = genre
        self.composer=composer
        self.language=language
        self.artist=artist
        self.songName=songName
```

This class has the goal to simplify the call of multiple criteria that the user choose when launching the program. This class is really important and is needed for every step of the program because it rely only on choosing criteria and to match the songs that fit the user's queries.

Now to open the plain text file we will use a for loop and the open function already present python. The program select the line in the text and split all the information when it finds a comma. After that, with the append method, the program will add the information in a blank list called here ”mySongs”. And finally with the called class it links information to parameters in the ”mySongs” list. Here is the loop and right after it an example of what is expected in the text file : For the rest of the code we use a while loop with the boolean value true such that :

```
for line in open("musics.txt"):
        startingLenght+=1
        line = line[0:-1]
        splitted = line.split(",")
        mySongs.append(Song(int(
        splitted[0]), splitted[1],
        splitted[2], splitted[3],
        splitted[4], splitted[5],
        splitted[6], splitted[7]))
```

For the criteria selection part, the two main key words used are the if statement and the for loop. Firstly, the program asks the user to activate or not the criteria by typing ”Y” to use that criteria and ”N” to not use that criteria to achieve the selection and to pass to another criteria. If the user wants to use that criteria, the program asks to give a value for that criteria. and all the songs that don’t match goes into a new empty list, and from that we delete all the songs in the ”mySongs” that does not fit the requirement.

```
requestScale= input
("Do you want to have the scale
criteria working on : ")
    if requestScale.lower() =="y":
        searchedScale = str(input("Enter the
        letter of the Scale you want : "))
        songs_to_remove = []
        for song in mySongs:
            if not song.scale == searchedScale :
                songs_to_remove.append(song)
        for song in songs_to_remove:
            mySongs.remove(song)
    for song in mySongs:
        print(song.path)
```

The end result of this is to have the ”mySongs” list that matches with the user’s requests. We do that for all the criteria to let the user choose. The program will finish by asking to give a name to the new audio file the user wants to export the final result in.

```
givenName = input("Enter the
name of file you want to export in : ")
    fileName = (givenName+".wav")
```

At the end of the code, the program verify if the starting length of the list equals the actual list of songs. If it is the same length the program start again due to the ”While True” loop. If the two list does not have the same length the program looks if there is no songs in the actual ”mySongs” loop, in the case there is no songs in the list the program starts again. In any other case the program display on the console that it as found ”i” songs, where i correspond to the length of the list. With a for loop, the display the list of songs that correspond to the user’s criteria. To finish, the program create a new varibale called ”finaltrack” that contains all the concatenated songs.

```
while True:
    //Various input for the criteria
    //execute or not the code with if
```

To open the song we will use the path object and the pydub keyword :

```
AudioSegment.from_wav(song.path)
```

6.4. Assessment (± 15% of section’s words)

cf. section **??** applied to the technical deliverable

Acknowledgment

The authors would like to thank the BiCS management and education team for the amazing work done.

7. Conclusion

The conclusion goes here.

References

[BiCS(2021)] BiCS Bachelor Semester Project Report Template. https://github.com/nicolasguelfi/lu.uni.course.bics.global University of Luxembourg, BiCS - Bachelor in Computer Science (2021).

[BiCS(2021)] Bachelor in Computer Science: BiCS Semester Projects Reference Document. Technical report, University of Luxembourg (2021)

[Armstrong and Green(2017)] J Scott Armstrong and Kesten C Green. Guidelines for science: Evidence and checklists. *Scholarly Commons*, pages 1–24, 2017. https://repository.upenn.edu/marketing_papers/181/

8. Appendix

All images and additional material go there.