Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma University

Pune – 411 048

Deployment Document

on

MEDIA PLAYER WITH INTERACTIVE SUBTITLE

Submitted By

Aksa Memon	**201700071(1780030023)**
Samyak Jain	**201700085(1780030010)**
Aqueel Kadri	**201700063(1780030015)**
Ayesha Memon	**201700070(1780030022)**
Jhanvi Pandya	**201700083(1780030035)**
Jyotiraditya Singh	**201700091(1780030043)**

School of Science and Technology
Year 2017 – 2021
BRACT's

Vishwakarma University
Survey No. 2/3/4 Kondhwa(Bk.) , Pune – 411048

Department of Computer Engineering

C E R T I F I C A T E

This is to certify that the project entitled

MEDIA PLAYER WITH INTERACTIVE SUBTITLE

Submitted by

Aksa Memon
Samyak Jain
Aqueel Kadri
Ayesha Memon
Jhanvi Pandya
Jyotiraditya Singh

is a bonafide work carried out by them under the supervision of Mr. S.J. Thaware and it is approved for the partial fulfillment of the requirement of Vishwakarma University for the award of the degree of Bachelor of Engineering (Computer Engineering) in the academic year 2017 -2021.

Mr. S.J. Thaware
(Project Guide)

(HOD.)

ABSTRACT

Over the last decade, human computer interaction has become an active research area, which releases people from inactive, inflexible communication with machines. A rich media service is a dynamic, interactive collection of multimedia data such as audio, video, graphics and subtitles.

The proposed solution of media player pertains to the intractability of subtitles. It implements the procedures of generating in a user interface an application window having a window frame and a plurality of panels within the frame. One panel of the said panels displaying subtitle statements with clickable functionality. Upon which the meaning of the respective word would be then displayed onto one of the said panels. The entirety of this process is done without the user having to switch any tab or without causing any inconvenience thereby helping the user comprehend the subtitles better and seamlessly.

INDEX

1) INTRODUCTION

Our goal is to make an efficient video player that not only plays the video and audio files smoothly and displays real time subtitles , but also has a feature which enables the user to search for meanings of the words which are being used in the video and present in the subtitles. The word to be searched is extracted from an inbuilt learning dictionary or an online dictionary along with an online search option.

While there are quite a few video players, only a small number of them facilitate the display of subtitles as and when the audio is played. Even fewer are able to sync the video with subtitles. This being one of the inconveniences, many a times the words that are used in the subtitles are not understood by a large number of audience.

Hence we aim to not merely sync up all three elements of the video, but also to allow the viewers to look up and understand the meanings of the dialouges in the video and hereby to make the video watching experience easy and comfortable for the user.

1.1) Purpose:

The purpose of our project is to develop a video player in such a manner that it will prove to be an interactive experience for the viewer. The video player will have subtitles playing along with the video which will improve comprehension. Our primary target is to make these subtitles reciprocative in nature which will aid the viewer in perceiving the meaning of the words and determine the interpretation of the video being played.

1.2) Scope:

- This media player is available for desktop operating systems such as Windows. While the platforms for this player could also be expanded to android and ios operating systems, the implementation of the same is a future scope of this project.
- Features: The project shall support audio and video files. It will play almost all video content. Some kinds of damaged, incomplete or unfinished videos can also run on this player.The UI of the media player shall include the media control icons such as play ( ), pause ( ), fast forward, back forward, and stop ( ) buttons.

- The current build of the project is available to read english subtitles. While the initial version of this player supports English language, it can also be upgraded to include other languages and their word dictionaries in order to increase the reach, to further ease the viewing experience and to serve the audience from around the globe including the non English speaking viewers.

- Formats:

  Input formats: The media player can read formats such as:

  Video Coding Formats : MP4, MKV.

  Subtitles format: SRT file.

2) Test Cases

ID	Case	Pre-condition	Steps	Result	Status
1	To check if the video playback is smooth and in good quality.	-	To play a high quality video in the player.	The video plays smoothly.	Pass
2	To check if the audio is clear and in sync with the video.	-	To play a high quality video in the player.	The video and audio are in sync and the sound of audio is clear.	Pass
3	To check if the subtitles are displayed at the right time.	Subtitle file is required.	To play a high quality video in the player.	The subtitle is in sync with video and audio.	Pass
4	If a word on the subtitles is clicked, the video should be paused and it's meaning should be displayed on the screen.	The subtitles should be there on the screen.	Click on the word(subtitle) to get the meaning displayed.	The meaning of the clicked word is displayed.	Pass
5	UI tools should be working properly like the play/pause/seek buttons.	-	Clicking on the play/pause/seek buttons.	When the play button is clicked the video will play.When pause button is clicked the video is paused.	Pass
6	Files of all supported formats need to be checked if they are playing properly.	Video-file, srt files are required.	Play the video in the player.	All the video files of various formats like MP4, MKV are working properly. srt file is also working properly.	Pass
7	When the user clicks on the subtitle part but in empty space.	The subtitles should be present on the screen.	Click between the words in the empty space.	Nothing will happen.The video will go on.	Pass

3) Project Major Module Pseudo Code

```
main()                  //entry point
{
        create bufferSubtitleClass object array of length 5;
        create panelWorkClass object;
        call miliClock.run function after every 1ms
}

bufferSubtitleClass
{
        string subtitleLine
        string startTime
        string endTime
}

miliClock Class
{
miliClock constructor()
{
                set subtitleBufferPointer
                set frameRate
                if beginning of the program
                {
                        store the 5 subtitles using the srt file into the bufferSubtitleClass-object array
                 }
```

```
}
void run()
{
        keep track of milliseconds passed (variable milli)
        variable str = (Convert milli to ms,seconds and hour format )
        if subtitleBuffer[subtitleBufferPointer].startTime is equal to str
        {
                print subtitle as buttons onto the screen(panelWork.addButtons)
                save the endTime of the subtitle
                fill the subtitleBuffer[subtitleBufferPointer] with the next subtitle
                update subtitleBufferPointer
        }
        if subtitleBuffer[subtitleBufferPointer].endTime is equal to endTime
        {
                Remove the subtitle buttons from the screen(panelWork.removeButtons)
       }
}
}

panelWork Class()
{
create frame   //along with necessary specifications for displaying the entire content onto the screen
create Panel   //along with the necessary specifications for displaying subtitles onto the screen
create panelForVideo  //along with the necessary specifications for displaying video image onto the screen
create keybindings and handle pause and play functionality(calling pauseAndPlayAction function)
add mouseMotionListener
display the pause and play icon onto the screen
```

```
play the audio                    //here created an independent thread
void printFrames()
{
        if the frameCount is less or equal to totalFrameCount
{
          store the image using frameGrabber into bufferedImage
          Update the background image of the video frame panel
}
}
void addButtons()
{
        split the subtitle string by the space regex
        add them into buttons in buttonList
        onClick Event should search the meaning of the word in the online and offline modes of
dictionary
        pause the screen and display the meaning onto the screen
}
void removeButtons()
{
    remove the subtitle buttons from the screen
}

void pauseAndPlayActions()
{
        if in Play mode
        {
                cancel the task scheduled onto the miliclockClass thread
```

```
                save the state of the miliClockObject
                suspend the Audio Player Thread
        }
            else
            {
                re-schedule the task from the previously saved state
                resume the Audio Player Thread
            }
}
}
```

4) Software Installation and Configuration

1) Java Development Kit

You run a self-installing executable file to unpack and install the JDK on Windows computers. Install JDK on Windows computers by performing the actions described in the following steps:

1.1) Downloading the JDK Installer

Access Java SE Downloads page and click Accept License Agreement. Under the download menu, click the download link that corresponds to the .exe for your version of Windows. JDK 8.2 and above recommended.
Link to download: https://www.oracle.com/java/technologies/javase-downloads.html

1.2) Installing the JDK Installer

You must have administrator privilege to install the JDK on Microsoft Windows.
To run the JDK installer:

1. Start the JDK installer by double-clicking the installer's icon or file name in the download location.
2. Follow the instructions provided by the Installation wizard.
3. After the installation is complete, delete the downloaded file to recover the disk space.

1.3) Setting the PATH Environment Variable

It is useful to set the PATH variable permanently for JDK so that it is persistent after rebooting. If you do not set the PATH variable, then you must specify the full path to the executable file every time that you run it.

To set the PATH variable on Microsoft Windows:

1. Select Control Panel and then System.
2. Click Advanced and then Environment Variables.
3. Add the location of the bin folder of the JDK installation to the PATH variable in System Variables.

2) JAR Files:

- ffmpeg
- ffmpeg-windows-x86_64
- javacpp
- javacpp-windows-x86_64
- javacv
- json-20140107
- opencv

5) FUTURE SCOPE

Our aim is to make a platform feasible to users and make the entire experience seamless so the users can enjoy the movies and other videos without interruption. While going forward our aim is to add more & more functionality and focus on user requirements. Which will enhance the user experience & give a holistic product that lets you do everything you possibly can inside the media player in the long run. The following are the very next steps that we are planning to accomplish in future.

1) Portability:

1.1) Multiple Subtitle Language: As we all know that subtitles are available in different languages like French, Spanish, German, Hindi, etc. So in the future we can integrate our software with these languages and not just English. Which will make our product more feasible to users.

1.2) OTT Platforms: This idea of interactive subtitles can also be integrated with OTT platforms like Netflix, Amazon, Hotstar, etc.

2) Feasible to access files:

2.1) Add Files by Drag and Drop: User can directly add those files which he wants to play by drag and drop with the mouse.

2.2) Add files by browsing: User will input the file by browsing the different directory or path of the file where it is located. The software automatically fetches the file from the given path.

3)Graphical Interface:

3.1) Menu bar: The menu bar will be displayed at the upper position of the media player where all the options regarding the software and the files will be given. For a particular task the user has to choose the proper option only. The user needs to click on the desired option as displayed to perform that specific task.

3.2) Status bar: The status bar is used for displaying the duration of the running file. Users can also seek in different positions of the file during run time through this status bar.

3.3) Fast forward: They can forward the media file with respect to the time when they want.

3.4) Volume control: It gives the facility to the user to control the volume of the video or audio file when it is running.

REFERENCES

1. http://bytedeco.org/javacv/apidocs/org/bytedeco/javacv/FFmpegFrameGrabber.html
2. https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html
3. https://docs.oracle.com/javase/tutorial/essential/concurrency/procthread.html
4. https://docs.oracle.com/javase/7/docs/api/java/util/TimerTask.html
5. https://ffmpeg.org/