

Implementation Programming language: JAVA

Library: JavaCV

Project Structure:

1. UI Class:

يحتوي كلاس الواجهة على واجهة البرنامج والتي تم استخدام JavaFx لبنائها والربط بينها وبين بقية الكلاسات وتتكون الواجهة من عدة ازرار وكل زر له وظيفة معينة (onAction method)

UI Class

Attributes:

main_stage(Stage)
player_video(Player)
Controller(MediaController)
Scroller(Scroll_image)
All_buttons(list of Buttons)

Methods:

UI():constructor
Scene GetUI():returns the GUI
Void ResetVideoTools(Player,BorderPane):resets the tools
Float NextValue(String):gets next value for the speed of fps
Void DisableButton(int[],Boolean):disables the buttons on certain condtion

2. File_Generate Class:

يحتوي هذا الكلاس على توابع للتعامل مع الملفات من انشاء وحذف وتعديل كما يحتوي توابع للمساعدة في عملية ال Undo

File_Generate Class

Attributes:

Files_temp(stack of files)
ID_Register (int)

Methods:

File CreateFile(String,String) : creates a file with specific suffix

void AddFile(File):Adds file to stack

File FileRemove(boolean):pops the last file in the stack

void Clear(): clears the stack

3. Player Class:

يحتوي هذا الكلاس على التوابع والخوارزمية الرئيسية لهذا المشروع ويتم عن طريقه انشاء objects من باقي الكلاسات لتنفيذ العمليات المرادة .

Player Class

Attributes:

Controller (MediaPlayer)

Video_MP4 (MediaView)

Grabber(FFmpegFrameGrabber)

CropRecorder(File)

ViedoWall(Pane)

Methods:

Player(File,Scene):Constructor

boolean Play_Pause(): play or pause the video

void SeekVideo(double): go to a specific frame in the video

void Volume_Setting(double): sets the volume of the video

void VideoTime(Slider): updates the slider timestamp

ArrayList<Image> Video2Images(): turns the video into frames

int GetFrameRate(): gets frame rate of the video in proper style

void ChangeSpeed(float):changes the fps rate

void CropVideo(int, int): crops the video from one point to another

boolean Paste2Vedio(Int):updates the view port after edit

ArrayList<Image> GetImage(File):returns array of video frames

Void TextWaterMark(): adds a text watermark to the video

void ImageWaterMark(): adds an image watermark to the video

void Merge(File , File): merges two video

4. Media Controller Class:

يحتوي هذا الكلاس على الاعدادت الخاصة بال media player للتحكم بها

Media Controller Class

Attributes:

TimeSeek (Slider)

Volume (Slider)

Player_video(Player)

Methods:

MediaController(Player) :Constructor

HBox ReturnTools(): returns all the media player tools

5. Scroll Image Class:

يحتوي هذا الكلاس على التوابع التي ستبني ال image scroller والتي ستعرض الفيديو على شكل frames ومنها يستطيع المستخدم ان يقوم بتعديل الفيديو from frame x to frame y

Scroll Image Class

Attributes:

ImagesV (list of ImageView)

ImageCache (list of ImageView)

From(int)

To(int)

Methods:

Scroll_Image(ArrayList<Image>):Constructor

void SettingImage(ArrayList<Image>):sets the images on the GUI

Integer GetFrom()

Integer GetTo()

void ResetObject(ArrayList<Image>):resets the images after an edit

6. Image2Mp4:

يحتوي هذا الكلاس على توابع تساعد على تحويل مجموعة من الصور الى فيديو مع امكانية ادخال صوت audio من ملف خارجي .

Media Controller Class

Attributes:

mp4SavePath (String)
audioPath (String)
Output(String)
imgMap(map of files)

Methods:

Void createMp4(String , Map<Integer, File>, int, int)
boolean mergeAudioAndVideo(String, String, String)

Solution Approach:

1. Video Crop:

- لكي يتم قص الفيديو بالطريقة الصحيحة نتخذ الخوارزمية التالية:
- I. نحول الفيديو الى صور frames بمساعدة FFmpegFrameGrabber ونخزن الصور داخل مصفوفة .
 - II. نختار ال frames المراد قصها من الفيديو ونحذفها من المصفوفة
 - III. نعيد بناء الفيديو من ال frames الموجودة داخل المصفوفة مع إضافة ال channels الخاصة بال audio كما تساعد مكتبة ffmpeg على الاحتفاظ بالصوت المرافق لكل frame من اجل تسهيل عملية ال video processing
 - IV. نحفظ الفيديو المعدل داخل مجلد temp
 - V. نعرض الفيديو المعدل على الواجهة.

2. Change fps:

- I. نستخدم ال FFmpegFrameGrabber ليعيد ال frames الخاصة بالفيديو
- II. نقوم بانشاء ملف فيديو فارغ جديد
- III. نسجل فيه ال frames الخاصة بالفيديو الاصلي ولكن نتعامل مع عملية ال iteration على ال frames بطريقة مختلفة حيث نمر على ال frames

ونضع حلقة داخل الحلقة الاصلية بحيث عندما يريد المستخدم التسريع فيتم تقليل مدة ظهور ال frame واذا كان يريد تخفيف السرعة فنسجل ال frame اكثر من مرة فيعطي وهم للمستخدم بان الفيديو يمشي ببطئ.

.IV نحتفظ بالفيديو الجديد داخل مجلد temp

.V نعرض الفيديو الجديد على الواجهة.

3. Adding watermark (image/Text) :

.I نستخدم ال FFmpegFrameGrabber ليعيد ال frames الخاصة بالفيديو

.II نقوم بانشاء ملف فيديو فارغ جديد

.III نمر على ال frames وعند كل frame نقوم باستخدام تابع frameFilter

الذي يساعد على وضع ال text/image على ال frame

.IV نحتفظ بالفيديو الجديد داخل مجلد temp

.V نعرض الفيديو الجديد على الواجهة.

4. Convert images to video:

.I نفتح المجلد الذي يحوي الصور ونقوم باخذ مسارات الصور عن طريق التابع

filelist ونخزنه داخل مصفوفة مكون من ال files

.II نقوم بترتيب المصفوفة حتى لا تواجهنا مشاكل ال flickering

.III نحدد ال framerate وال encoding type ولاحقة الفيديو المراد انشائه

.IV نمر على جميع الملفات ونستخدم التابع Java2DFrameConverter ليحول

الصور ال frame

.V نسجل كل frame بشكل متتالي عن طريق ال recorder

.VI نحفظ الفيديو الجديد في المسار المحدد مسبقا.

5. Adding audio to the video:

.I نحدد مسار الفيديو والصوت

.II نستخدم التابع ال FFmpegFrameGrabber لجلب ال frames الخاصة بكل

من الفيديو getframe والصوت get audio channels.

.III نحدد ال frame rate لكل من الفيديو والصوت ليتناسق الخرج

.IV نمر على video frames ونسجلها عن طريق ال recorder

.V نمر على audio frames ونسجلها عن طريق نفس ال recorder

.VI نحفظ الفيديو الجديد في المسار المحدد.

6. Merging two videos:

- I. نحدد مسار الفيديو الاول
- II. نحدد مسار الفيديو الثاني
- III. نستخدم تابع ال FFmpegFrameGrabber لإنشاء Grapper لكل فيديو
- IV. نقوم بإنشاء فيديو فارغ جديد
- V. نمر على الفيديو الاول ونقوم بتسجيل جميع ال frames الخاصة به عن طريق ال recorder
- VI. نمر على الفيديو الاول ونقوم بتسجيل جميع ال frames الخاصة به عن طريق ال recorder
- VII. نحفظ الفيديو الجديد داخل مجلد temp
- VIII. نعرض الفيديو الجديد على الواجهة

7. Undo operation:

- I. نقوم بالبحث عن اخر فيديو تم حفظه في مجلد ال temp عن طريق التابع LastFile
- II. نقوم بعمل reset لل media player
- III. نحذف الملف من مجلد ال temp
- IV. نعرض الفيديو على الواجهة.

Results:

تم انجاز ما يلي :

1. إمكانية ادخال ملف فيديو الى التطبيق.
2. عرض الفيديو المدخل على شكل تتالي لمجموعة صور منه أثناء تحريره.
3. قص/ حذف مقطع معين من الفيديو.
4. إمكانية تحريك (تغيير مكان) مقطع معين من الفيديو.
5. إضافة علامة مائية watermark ان كانت صورة | نص.
6. حفظ الفيديو الناتج بعد تحريره كملف على القرص الصلب.
7. إمكانية تغيير عدد frames بالثانية (FPS) أثناء حفظ الفيديو.
8. إمكانية التراجع عن العملية الأخيرة التي قام بها المستخدم أثناء تحرير الفيديو.
9. إمكانية إنشاء فيديو جديد انطلاقاً من مجموعة صور متتالية (مع او بدون صوت)
10. بشكل متتالي زمنياً إمكانية دمج أكثر من مقطع فيديو معاً.
11. تنفيذ عمليات تحرير الفيديو بزمان مقبول حتى على الفيديوهات الطويلة نسبياً

الواجهة:

