



**Ciências
ULisboa**

**Departamento de Informática da Faculdade de
Ciências da Universidade de Lisboa**

Multimédia 2022/2023

Projeto Final

Atmosvideo

“Gera música ambiente para os teus vídeos”

Grupo 06

Aluno	Número
José Siopa	fc60716
Miguel Roldão	fc57485

10 de julho de 2023

ÍNDICE

1	INTRODUÇÃO.....	1
2	TRABALHO RELACIONADO.....	2
3	EXTRAÇÃO DAS PROPRIEDADES DE VIDEO	2
3.1	Implementação	3
3.1.1	Cálculo da Energia	3
4	GERAÇÃO DA MÚSICA.....	4
4.1	Sintetizador.....	4
4.1.1	Interface com o sintetizador	4
4.1.2	Soundfont utilizada	4
4.2	Propriedades do vídeo afetam a geração da música	5
4.2.1	Tempo	5
4.2.2	Transposição	5
4.2.3	Escala	5
4.2.4	Tipo de Acordes	6
4.2.5	Instrumentos.....	7
4.3	Gerar notas musicais	7
5	INTERFACE GRÁFICA	8
6	TRABALHO FUTURO.....	9

1 Introdução

Para este projeto, vamos construir uma aplicação que se engloba no tema de processamento e visualização multimédia.

Visto que o vídeo é um meio de informação muito rico, este vai ser utilizado como base da aplicação. O utilizador vai poder escolher um vídeo e a aplicação irá gerar música ao decorrer deste, conforme as variações dos níveis de energia e de cor detetadas ao longo do vídeo.

Os objetivos a cumprir para realizar esta aplicação são os seguintes:

- Carregar vídeos à escolha do utilizador ou escolher de entre vídeos já existentes na aplicação;
- Extrair informações pertinentes do vídeo escolhido, tais como movimento, matriz de cor, saturação, luminosidade e som do vídeo original;
- Gerar música para o vídeo a partir de uma interpretação das informações extraídas ao longo do vídeo;
- Implementar uma interface onde o utilizador possa interagir com o vídeo (pausar, avançar, recuar, alterar o volume);

Para a implementação do projeto, vamos ter de recorrer a diversas bibliotecas para conseguirmos obter o produto final. Vamos utilizar o OpenCV para extrair informações sobre os pixéis do vídeo, o FluidSynth para gerar música programaticamente e o CustomTkinter que permite criar uma interface gráfica com maior facilidade.

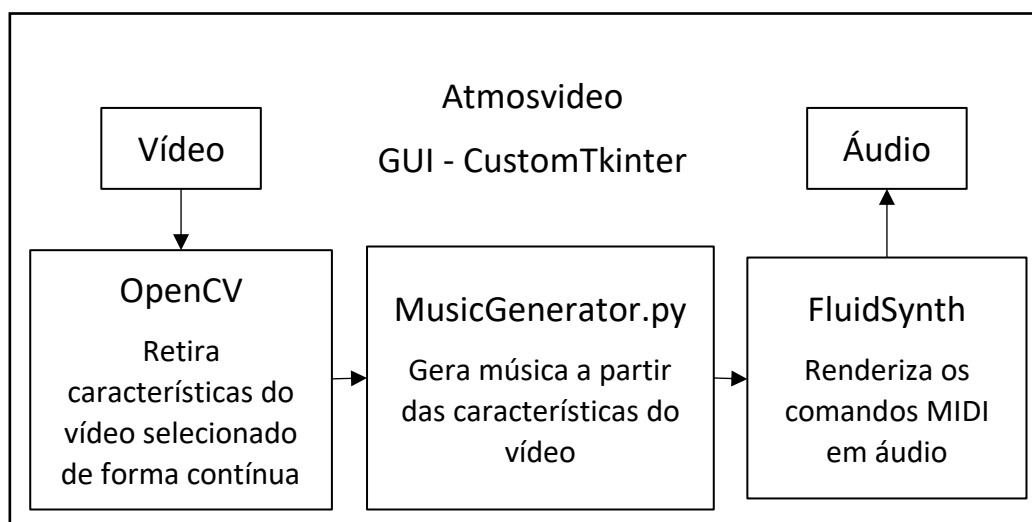


Figura 1 – Atmosvideo pipeline

2 Trabalho Relacionado

1. Margounakis, D., and Politis, D. 2006. *Converting Images to Music Using their Color Properties*. In *Proc. of the 12 th Int. Conf. on Auditory Display*, London, UK, Jun.

Este artigo discute a antiga associação entre música e cores, e como diferentes mapeamentos entre som e imagem podem permitir uma conversão eficiente entre os dois. É descrito um método proposto para conversão de imagens em música usando o conceito de cromatismo, que fornece um ambiente parametrizado para apresentações audiovisuais. A exibição auditiva de imagens coloridas pode revelar diferentes maneiras pelas quais um ouvinte percebe uma peça musical devido às transições de cores. O artigo também descreve um modelo de design para síntese cromática e apresenta um pequeno exemplo baseado em um ícone gráfico digital para demonstrar resultados preliminares.

2. *MetaSynth 5: Electronic music studio and sound designer's dream tool for Mac OS X*

Esta é uma aplicação já existente que permite programar música eletrônica através de pinturas simples realizadas pelo utilizador com base nas cores e no brilho destas.

3. *The vOICE: Augmented Reality for the Totally Blind*

Esta aplicação permite ajudar pessoas com problemas visuais ao traduzir o que a camera captura em sons, e ao usar o pitch e stereo pan, um utilizador treinado pode conseguir identificar que objetos são e onde se encontram.

4. Teresa Chambel, Sérgio Neves, Celso Sousa and Rafael Francisco. *Synesthetic Video: Hearing Colors, Seeing Sounds*. *MindTrek 2010, October 6th - October 8th, 2010, Tampere, FINLAND*

Este artigo consiste num vídeo interativo que permite ouvir as cores e influenciar as propriedades visuais através do som e da música recorrendo a interações do ambiente e do utilizador, por exemplo, com uma lanterna.

3 Extração das propriedades de video

Analizou-se que tipos de propriedades poderiam ser extraídas de um video e que fossem uteis para a geração de música. A primeira que foi escolhida como a mais importante é a energia. Esta propriedade refere-se ao movimento entre frames de um video, as diferenças entre frames. Outras propriedades que considerámos relevantes e computacionalmente faceis de obter, são a cor do video, o brilho e a saturação de cor. Ou seja, os valores *hue*, *saturation* e *value* do modelo de cor HSV.

Para a extração, manipulação e cálculos realizados com frames optou-se por utilizar a biblioteca OpenCV. Pela sua performance e por ser open source.

3.1 Implementação

A implementação da extração destes parâmetros revelou-se um desafio. Pois devido à relativamente baixa velocidade de execução da linguagem python, as primeiras iterações desta componente eram realmente vagarosas, chegando a processar apenas 10 frames por segundo na máquina de referência. A implementação final consegue processar, em média, cerca de 140 frames por segundo. Obtendo um aumento de velocidade em 14 vezes. A pipeline atual é ilustrada na Figura 2.

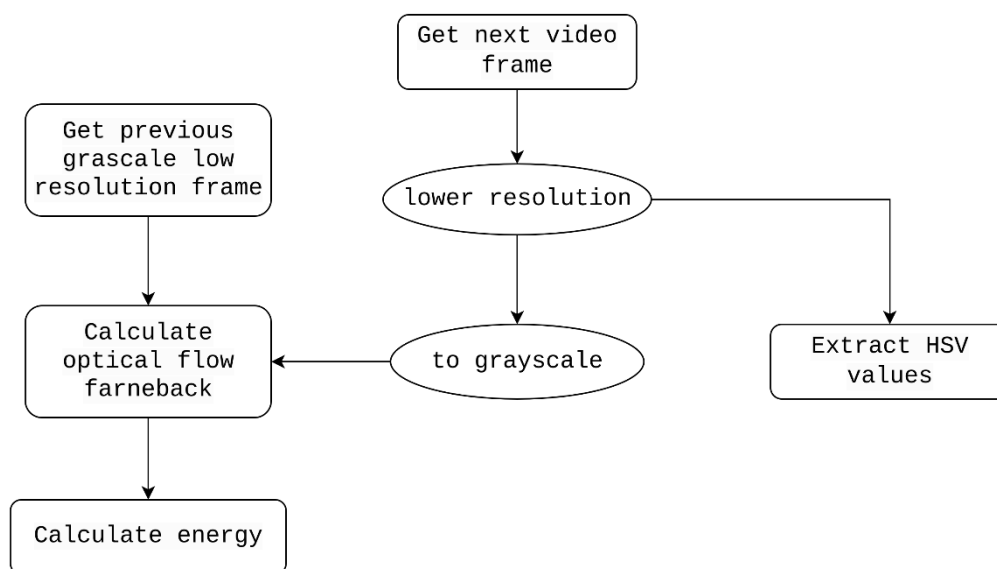


Figura 2 – Pipeline da extração dos parâmetros do vídeo

A pipeline começa por ler a próxima frame de vídeo e redimensiona-a para uma resolução mais pequena, 180p. Com esta nova resolução reduz-se drasticamente o número de píxeis a ser processados para a extração dos parâmetros. Esta frame reduzida, é ainda convertida para escala de cinzentos. Reduzindo as componentes de cor de 3 para 1. Que é o minimamente necessário para o cálculo da energia. A partir daqui o programa separa-se em várias threads. A primeira, calcula os valores HSV da nova frame utilizando a frame reduzida a cores. As restantes (cujo número depende do número de processadores presente no computador), extraem o valor da energia.

3.1.1 Cálculo da Energia

O principal bolo do cálculo da energia é o cálculo de vectores de movimento interframe. Estes vectores relacionam os conteúdos de duas frames, e representam a distância e direcção em que cada pixel se tem que movimentar para a passagem de uma frame para a outra.

Para o efeito, utilizamos a função do OpenCV `calcOpticalFlowFarneback()` que vai calcular precisamente os vectores de movimento. Cada thread chama esta função apenas sobre os píxeis que lhe correspondem.

Posteriormente, calcula-se a magnitude de cada um destes vectores. A média desta magnitude entre todos os pixels de todas as threads é o valor da energia. Para evitar picos de energia desproporcionais ao resto do vídeo, limitamos este valor a 1.0.

4 Geração da Música

Para gerar a música foram definidos 2 instrumentos, um que toca acordes e um que toca a parte da melodia. Estes instrumentos geram música processualmente, a partir de números aleatórios, atendendo a certos parâmetros musicais.

4.1 Sintetizador

As notas geradas são então enviadas para um sintetizador MIDI, que gera o som. Escolhemos o sintetizador FluydSynth, por ser um sintetizador *opensource* e ter uma API em *python* bastante simples de utilizar. Optamos por utilizar um sintetizador MIDI, porque este nos permite utilizar uma vasta quantidade de instrumentos, este adiciona um nível de abstração no que toca à geração de som por ser um standard na indústria musical para o efeito.

4.1.1 Interface com o sintetizador

A API em *python* do “FluydSynth” facilitou bastante a comunicação entre este e a nossa aplicação.

Para a inicialização do sintetizador, apenas precisamos especificar: o ganho, que afeta o volume geral do sintetizador. Escolhemos um valor de 2, para igualar o volume do sintetizador a outras aplicações; e a frequência de amostragem. Aqui deixamos o valor padrão de 44100 Hz, que um valor comumente utilizado e equivale a qualidade de áudio CD.

Para controlar as notas, utilizamos duas funções que correspondem aos comandos MIDI “note_on” e “note_off”. Onde precisamos especificar o canal MIDI, o número da nota em MIDI, e a velocidade (i.e., o volume). Para o canal MIDI, atribuiu-se o canal 0 ao instrumento dos acordes e o canal 1 ao instrumento da melodia. O número da nota MIDI é um inteiro entre 0 e 127 e varia em frequência de 8.18 Hz até 13.29 kHz. Este número depende da nota gerada por cada instrumento. À velocidade da nota dá-se o valor do volume do instrumento utilizado.

4.1.2 Soundfont utilizada

Para o sintetizador ser capaz de gerar áudio, este necessita de uma especificação dos sons a utilizar. Para esta especificação, o sintetizador utiliza um ficheiro SoundFont que contem uma serie de instrumentos com vários trechos de áudio gravados em amostras PCM, de forma a poder tocá-los na chegada de comandos “note_on”.

Para o efeito, escolhemos a soundfont ColomboGMGS2, pela variedade de instrumentos que contem e pela sua utilização se encontrar sob a licença “Creative Commons”.

4.2 Propriedades do vídeo afetam a geração da música

Para que a música gerada seja adequada ao vídeo, foi necessário um planejamento de como cada parâmetro da geração da música é afetado pelas várias propriedades extraídas do vídeo. Na implementação apresentada, cada parâmetro é alterado por uma propriedade (ou um conjunto de propriedades) do vídeo. No entanto, se estes estiverem diretamente ligados, a variação constante dos parâmetros resulta numa música instável e incoerente.

De modo a resolver este problema, os parâmetros são apenas atualizados após uma variação considerável das propriedades do vídeo. Ou seja, quando a diferença do novo valor da propriedade e do último valor utilizado é maior que X. Através de experimentação, demos a X um valor de 10% do último valor utilizado.

4.2.1 Tempo

O tempo da música é determinado apenas pelo nível de energia, varia linearmente entre 50 BPM e 160 BPM. Este é um intervalo aceitável que permite a geração de músicas lentas e rápidas. Este parâmetro só é alterado se o tempo da música for mais ou menos 20% que o tempo atual.

4.2.2 Transposição

Este parâmetro é útil para controlar a frequência das notas escolhidas pelos instrumentos. Com um valor mais baixo as notas são mais graves, para um valor mais elevado as notas são mais agudas. Este parâmetro é determinado apenas pela propriedade “value”.

Para o instrumento da melodia este parâmetro varia entre -1 e 2. Ou seja, as notas geradas podem ser desde uma oitava abaixo até duas oitavas acima do valor normal. Assim o instrumento de melodia pode servir como baixo em vídeos com uma propriedade “value” mais reduzida.

Já para o instrumento dos acordes, este parâmetro varia da seguinte forma:

$$\begin{aligned} \text{if } value < 0.5: \text{transposition} &= 1 - 2 \cdot value \\ \text{else: } \text{transposition} &= 2 \cdot value - 1 \end{aligned}$$

Deste modo, os acordes utilizam notas agudas para valores altos de “value” altos e baixos. Mas para notas médias, as notas dos acordes são mais graves.

Com os dois instrumentos combinados, podemos perceber a utilidade deste parâmetro. Para valores de “value” baixos, o instrumento da melodia toca uma linha de baixo, enquanto os acordes tocam as notas mais agudas; para valores médios, a melodia toca mais aguda que os acordes. O mais comum numa música; Para valores altos, os acordes são agudos e a melodia ainda mais aguda. Criando uma atmosfera que evoca uma sensação de estar a sonhar, sem notas graves.

4.2.3 Escala

A escala utilizada para escolher as notas geradas depende apenas da propriedade “hue”, como ilustrado na Figura 3.

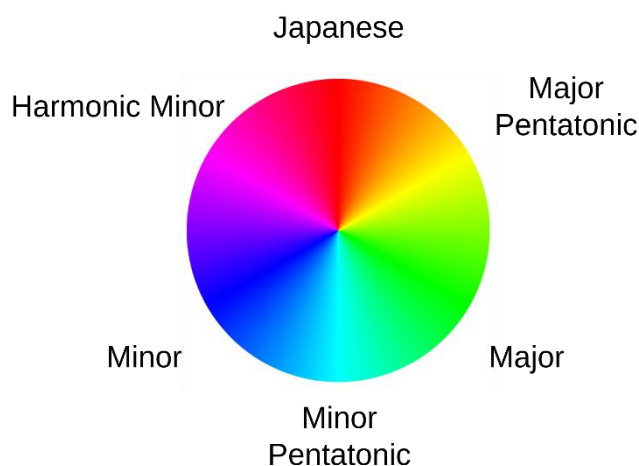


Figura 3 – Associação das escalas às cores do vídeo.

Para chegar a estas associações entre cor e escala, correlacionou-se as emoções que certas cores podem provocar no ser humano com as emoções que as escalas musicais transmitem.

Começando pelas escalas mais comumente associadas a emoções, a escala maior e a escala menor, geralmente associadas com o sentimento de alegria e tristeza respetivamente. Foram-lhes atribuídas as cores verde e azul. No entanto a cor mais associada à alegria é tipicamente o amarelo. Decidimos associar esta cor a escala maior pentatónica, por esta transmitir o mesmo sentimento de alegria com a diferença de ter mais energia.

A escala menor pentatónica, é um bom ponto intermédio entre a escala maior e a menor. Foi-lhe então atribuída a cor ciano.

Tendo ainda as escalas japonesa e a escala menor harmónica, atribuímos à primeira, a cor vermelha. Por estar associada a paixão e energia. E à escala menor harmónica foi-lhe atribuída a cor magenta, pelas suas propriedades sonoras dissonantes e inquietantes.

4.2.4 Tipo de Acordes

Foram implementados 3 tipos de acordes:

- Power chords: Compostos por apenas duas notas. São muito utilizados em músicas mais energéticas como o blues e o rock. Utilizam a primeira e a quinta notada escala.
- Triads: Compostos por três notas. São os acordes mais comumente utilizados na maior parte dos estilos musicais. Utilizam a primeira, a terceira e a quinta nota da escala.
- Seventh chords: Compostos por quatro notas. São acordes mais complexos, especialmente utilizados em jazz e secções mais atmosféricas. Utilizam a primeira, a terceira, a quinta e a sétima nota da escala.

Para decidir qual o tipo de acordes que vai ser tocado, calcula-se a média entre a propriedade “saturation” e o inverso da energia. Este valor equivale à complexidade do

acorde. Se este valor for baixo, opta-se pelo “power chord”; se o valor for mediano, o “triad”; e se o valor for alto, opta-se pelo “seventh-chord”.

4.2.5 Instrumentos

Para a escolha dos instrumentos, achamos que faria o mais sentido serem decididos pela energia e pela propriedade do vídeo “value”. Os instrumentos foram escolhidos por experimentação e gosto pessoal.

4.3 Gerar notas musicais

A geração das notas musicais é feita, em grande parte, aleatoriamente. Cada vez que uma nota é gerada, gera-se um número aleatório entre 0 e 1.0. Esta variação equivale a uma oitava. A este número, é-lhe adicionado o parâmetro da transposição. Se este for parâmetro for positivo, a nota final fica mais aguda, se for negativo, a nota fica mais grave.

Este valor, juntamente com a escala ativa, permite à função `note_from_scale()` escolher uma nota sonante quando tocada juntamente com outros instrumentos na mesma escala.

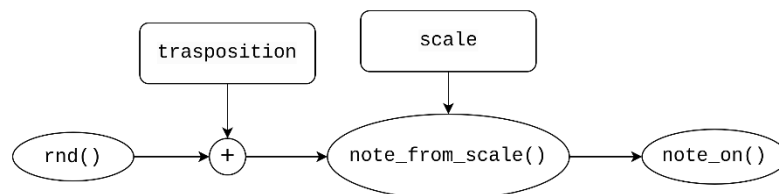


Figura 4 – Geração de notas para o instrumento da melodia.

Assim é feita a geração de notas para o instrumento da melodia, sendo que ainda existe uma probabilidade de 20% de o instrumento fazer uma pausa em vez de tocar uma nota.

O processo para o instrumento de acordes é semelhante, à diferença que este trabalha com várias notas ao mesmo tempo.

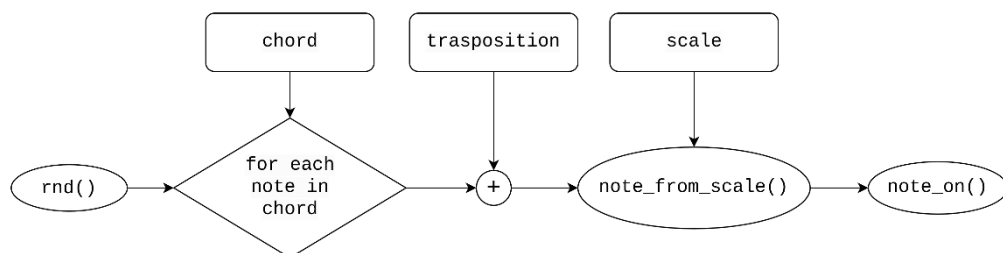


Figura 5 – Geração de notas para o instrumento dos acordes.

O instrumento de acordes utiliza um dos tipos de acordes definidos com informação à cerca dos intervalos entre as notas a utilizar. Na Figura 6 podemos observar um exemplo de como este instrumento seleciona as notas tendo como parâmetros: um acorde do tipo *triad*, o valor de 2 para o modo (que é calculado de acordo com a Equação 1), utilizando a escala maior, com base na nota dó, e com a transposição a 0.

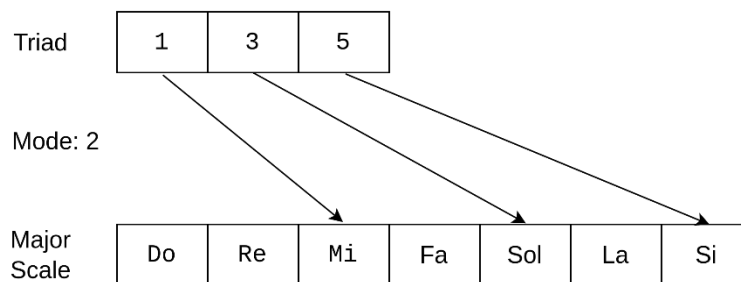


Figura 6 – Exemplo da escolha de notas com um acorde.

```
mode = floor(rnd() * number_of_notes_in_scale)
```

Equação 1 – Cálculo do valor “modo”.

O instrumento dos acordes pode ainda funcionar em modo *arpeggio*, onde as notas do acorde são tocadas sequencialmente, em vez de todas ao mesmo tempo.

5 Interface Gráfica

Para a criação da interface gráfica, optou-se por utilizar a biblioteca CustomTkinter. Esta permite facilmente criar interfaces, onde cada componente é uma classe no python com diferentes propriedades alteráveis.

Viste que a aplicação se foca no áudio, decidiu-se utilizar uma interface simples e pouco distrativa, para que o utilizador se sinta imerso ao visualizar o vídeo. Seguindo o plano previamente definido, colocou-se o vídeo no topo esquerdo com os controlos dele por baixo e no lado direito, uma lista de vídeos já existentes na pasta do projeto.

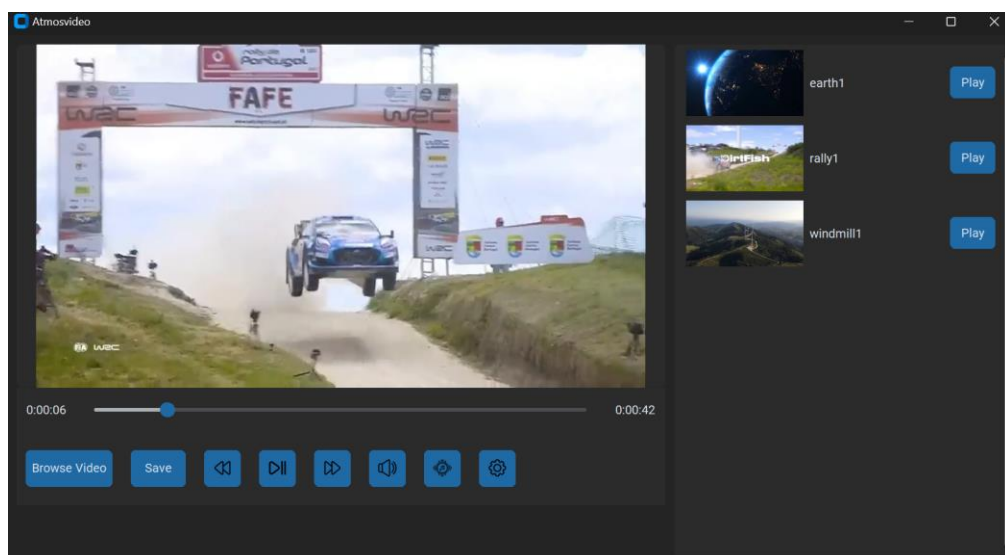


Figura 7 - Interface Principal

Para colocar um vídeo para ser reproduzido, existe a opção de escolher um vídeo em qualquer lugar do computador do utilizador ou clicando num dos que estão na lista à direita. Depois, é possível pausar e resumir o vídeo, alterar o volume, avançar ou recuar e há um botão que irá gerar áudio para o vídeo atual. Para o reprodutor de vídeos, utilizou-

se a biblioteca vlc, que por sua vez requer ter o programa para o funcionamento da aplicação.

Quando se clica no botão para gerar áudio, o utilizador passa para uma tela de espera enquanto a aplicação extrai as propriedades do vídeo e gera o áudio, utilizando as classes referidas anteriormente. O tempo de espera varia conforme o comprimento e complexidade do vídeo escolhido.

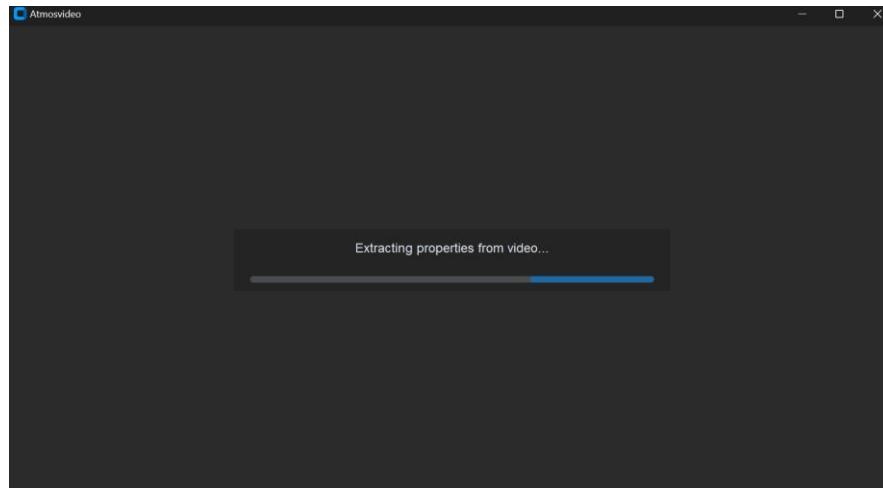


Figura 8 - Tela de espera enquanto é gerado o áudio para o vídeo

Depois, é criado um ficheiro temporário WAV e uma cópia temporária do vídeo e o áudio é adicionado a este, utilizando a biblioteca moviepy. O ficheiro WAV é descartado e o novo vídeo temporário é carregado para o reproduutor. Para manter o vídeo, basta o utilizador guardar o vídeo na pasta vídeos do projeto, onde é gerada uma thumbnail e o vídeo é adicionado na lista do lado direito da aplicação.

6 Trabalho futuro

No que toca à geração de música, a escolha dos instrumentos poderia ser feita pelo utilizador através da interface gráfica. Uma vez que não achamos nenhuma correlação lógica entre os parâmetros extraídos e o tipo de instrumento acabamos por os atribuir por gosto pessoal. A escolha manual do nível de energia de determinadas zonas do vídeo daria mais liberdade e controlo ao utilizador para alcançar a música desejada, mas aumentaria a complexidade da aplicação.

A própria música gerada acaba por soar robótica e artificial. Uma possível causa deste efeito é as músicas serem demasiado aleatórias. A utilização de sequências de notas estudadas e predefinidas poderia reduzir este efeito. Ou a repetição de certos segmentos, como também é comum na música.