

GENERACIÓN AUTOMÁTICA DE MÚSICA GUIADA POR LA ARMONÍA DEL COLOR

AUTOR:
CARLOS COMPANYY BERNAL

TUTOR:
CARLOS PÉREZ SANCHO

DEPARTAMENTO:
LENGUAJES Y SISTEMAS INFORMÁTICOS

OCTAVO CURSO

Índice

1. Introducción	3
2. Módulo de extracción de datos de colores.....	3
2.1 Simplificación de la imagen.....	3
2.2 Extracción de datos.....	4
2.3 Ejemplos de extracción de características de una imagen.....	5
3. Correspondencia entre colores y música.....	7
3.1 Asociaciones.....	8
3.2 Tabla resumen de parámetros.....	9
4. Módulo compositor musical automático.....	9
4.1 Cálculo de las probabilidades de duración de nota en base a la velocidad.....	9
4.1.1 Ejemplos de notas generadas aleatoriamente usando los pesos de velocidades concretas.....	10
4.2 Cálculo del tempo en base a la velocidad	11
4.3 Generación de música automática.....	11
4.3.1 Normas de composición básicas.....	11
4.3.2 Normas de composición adicionales para la progresión de acordes.....	11
4.3.3 Normas de composición adicionales para la melodía.....	12
5. Dificultades encontradas en el proyecto.....	12
6. Manual de usuario (interfaz).....	13
7. Bibliografía y enlaces de interés.....	15

1. Introducción

Este proyecto consiste en la generación de música de forma automática guiada (condicionada) por la armonía del color de una paleta de colores o una imagen.

Tendremos por lo tanto dos desafíos muy diferentes. Por un lado tendremos que ser capaces de analizar imágenes y datos sobre colores y conseguir extraer información suficientemente relevante de ellos que nos proporcione unos resultados que diferencien e identifiquen cada caso de una forma clara. Por otro lado tendremos que desarrollar un algoritmo que genere música coherentemente en base a unos parámetros, con todas las dificultades que ello conlleva.

Algunas de las motivaciones para el desarrollo de proyectos de este tipo pueden ser el desafío o la mera curiosidad por intentar automatizar y codificar en reglas un proceso tan creativo y basado en la subjetividad como es la creación de música. Otras más prácticas y ambiciosas pueden ser por ejemplo la integración de personas con deficiencias visuales a través de la transformación de los colores en sonidos, de manera que puedan aprender a escuchar lo que no pueden ver.

Hay que decir que se pueden encontrar a lo largo de internet muchos intentos de generar música mediante algoritmos e incluso algún ejemplo de algo parecido a lo que aquí estamos intentando desarrollar, pero no hay nada satisfactorio destacable en cuanto a la generación de música automática que hayamos podido encontrar debido a las dificultades que conlleva el intentar aunar la creación artística y la automatización.

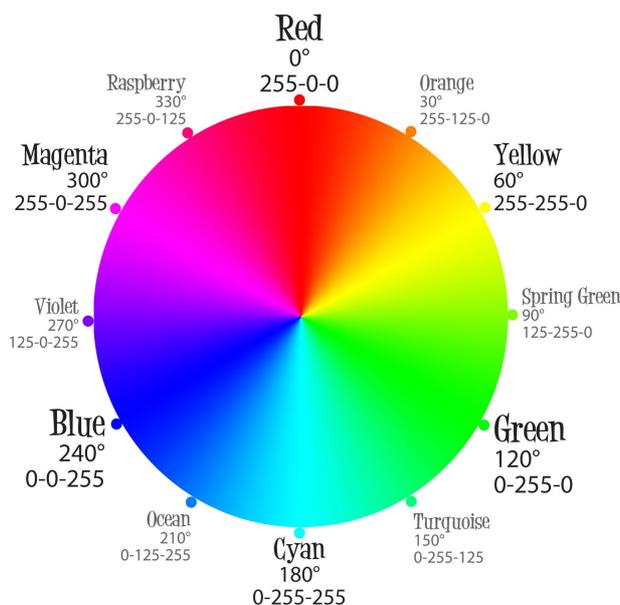
Podemos separar las secciones o módulos del proyecto en tres partes. La primera será la que se encargue de extraer los datos relevantes de una imagen o una paleta de colores. La segunda establecerá una correspondencia entre los datos que tienen que ver con los colores y los datos que tienen que ver con la música, es decir, convertirá los parámetros de salida del primer módulo, en parámetros de entrada del compositor musical automático. El tercer módulo será el compositor que generará música de forma automática y continua en base a los parámetros de entrada que reciba. A continuación explicaremos por separado cada uno de estos módulos.

2. Módulo de extracción de datos de colores

Este módulo funciona prácticamente igual tanto si los datos a extraer provienen de una imagen o de la paleta de colores, ya que cada cuadro de elección de la paleta de colores acaba siendo equivalente a lo que es un píxel de la imagen, con lo que a partir de ahora explicaremos todo lo relativo a la extracción de datos de una imagen sin hacer referencia a como se haría para la paleta de colores ya que son equivalentes.

2.1 Simplificación de la imagen

Lo primero que debemos hacer es reducir el número de colores que tiene la imagen. Esto lo haremos debido a que para extraer las características relevantes de la imagen debemos poder cuantificar el porcentaje que hay de cada color en la imagen y para esto necesitamos agrupar los colores en grupos básicos, en lugar de trabajar con los millones de colores que existen. De este modo, los colores básicos con los que trabajaremos serán los doce del círculo cromático:

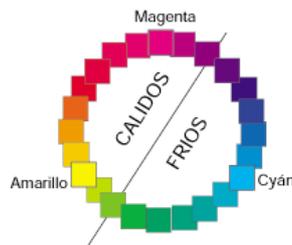


Para reducir los colores de una imagen a los doce básicos con los que vamos a trabajar, lo que haremos será simplemente calcular para cada píxel de la imagen cual de los doce colores básicos es el que más se acerca a él. Consideramos que la distancia entre dos colores es la suma de cada una de las diferencias que hay entre sus respectivos valores R,G,B. Como nota hay que decir que aparte de los doce colores del círculo cromático también contamos de cara a aproximar el píxel hacia ese color a los colores blanco y negro, descartando esos píxeles, o dicho de otro modo, los píxeles que el color al que más se aproximen sea el blanco o el negro serán simplemente

descartados. Esto puede provocar que ciertas imágenes en tonos oscuros puedan dar resultados extraños a la hora por ejemplo de establecer el color predominante, que visualmente puede parecer uno pero a la hora de aproximar muchos de esos tonos de por ejemplo azul oscuro irán a parar al negro y serán descartados, dejando a otros colores aparentemente menos presentes con mayor porcentaje real de presencia en la imagen. De todas formas se podrá apreciar con ejemplos que no es algo que desvirtúe los resultados en la mayoría de las imágenes.

2.2 Extracción de datos

Una vez que tenemos la imagen reducida a doce colores, el primer dato relevante que obtendremos es el porcentaje que hay de cada uno de los colores en la imagen. Con estos porcentajes podremos obtener en primer lugar el color mayoritario de la imagen (el que se encuentra en mayor porcentaje). También obtendremos el porcentaje de colores cálidos y fríos considerando como colores cálidos los que van desde el Magenta al Spring Green y como fríos desde el Turquoise al Violet:



El último dato relevante que extraeremos de la imagen es el nivel de policromismo de la misma. Por policromismo consideraremos la cantidad de colores diferentes que tiene la imagen, de manera que cuantos mas colores relevantes tenga mas alto sera su valor de policromismo y viceversa (en realidad será la cantidad de colores que se desvían de la cantidad que nosotros consideramos como normal o básica). Esto es mas complicado de lo que parece, ya que necesitamos acotar el rango de valores que puede tener esta variable y al mismo tiempo necesitamos que la manera de calcular esos valores genere resultados que se distribuyan uniformemente y de manera lógica para las diferentes imágenes. Dicho de otro modo, la dificultad radica en primer lugar en establecer cuando consideramos que un color es relevante en la imagen (a partir de que porcentaje de presencia en la imagen se considera relevante), como establecer el número de colores que se consideran como cantidad normal para sobrepasar un determinado porcentaje (para establecer a partir de ahí una desviación de la cantidad normal de ese número de colores que se considera normal y poder extraer a partir de ahí un valor de policromismo que será considerado como el porcentaje de desviación de la cantidad de colores que esperamos) y conseguir que además esto genere valores acotados en un rango (por ejemplo de cero a cien) y que las diferentes imágenes generen valores que abarquen todo ese rango. Basándonos en la observación y en la realización de pruebas sobre un alto número de imágenes, llegamos a unos valores que satisfacen esas necesidades y generan unos resultados lógicos y coherentes que mas adelante mostraremos con ejemplos sobre imágenes reales.

Los valores que calcularemos serán:

- Número de colores que superan el 3% de presencia. Considerando como valor normal, o dicho de otro modo valor con desviación cero, dos colores. A partir de ahí y hasta diez colores máximo se considerará como desviación, siendo diez colores considerados como una desviación del 100%. Se puede pensar que el valor máximo de desviación debería producirse con doce colores (ya que en realidad es el número máximo de colores con el que trabajamos), pero si hiciéramos esto estaríamos disminuyendo el porcentaje de desviación que produce cada color que encontramos y como mediante la observación hemos encontrado que prácticamente no hay imágenes que tengan doce colores que superen el 3% de presencia, es mejor utilizar el porcentaje de desviación con respecto a ocho colores (de dos a diez), ya que cuando encontremos un color mas que supere el 3% repercutirá más en el porcentaje de desviación que si pudiera haber hasta doce. Esto hace que perdamos algo de precisión por el lado de que una imagen que tenga diez colores que superen el 3% va a dar el mismo valor de desviación que una imagen que tenga doce valores que lo superen, pero por otro lado hace que ganemos precisión para un grupo de imágenes que no tengan mas de diez valores que superen el 3% de presencia.
- Número de colores que superan el 0.5% de presencia considerando como valor normal o desviación cero cuatro colores, y considerando el máximo numero de colores que pueden superar el 0.5% diez. Por las mismas razones que en el caso anterior.
- Número de colores que superan el 0% de presencia considerando como valor normal ocho y considerando como valor máximo posible de colores que superen el 0% once.

Vemos que hemos realizado los cálculos a tres niveles diferentes así que luego tendremos que sumarlos y dividirlos entre tres para obtener el porcentaje de policromismo final. La razón de realizar los cálculos a tres niveles diferentes es debido a que por separado (utilizando por ejemplo solo el cálculo de colores que superan el 3%) había imágenes que no quedaban bien evaluadas ya que quizá solo tenían cuatro colores que superaran el 3% pero luego tenían muchos que superaban el 1% y luego podía haber otra imagen con cuatro colores también que superaran el 3% y que luego el resto no superaran apenas el 0%. Estas dos imágenes habrían dado el mismo valor de policromismo siendo esto poco coherente. De la misma manera, la solución no era bajar el porcentaje que usamos para considerar un color como relevante por ejemplo al 1%, ya que también se nos escapa la medición por otro lado al poder haber una imagen en la que superen diez colores el 1% de presencia pero que luego haya dos mayoritarios que se repartan casi todo el porcentaje, la cual daría el mismo valor de policromismo que una imagen que tiene diez colores que superan el 1% de presencia y en la que en cambio si tienen todos un buen porcentaje de reparto final de presencia. Este método consigue medir de una forma eficaz el nivel de policromismo de una imagen y como veremos a continuación genera unos resultados bastante lógicos y que consiguen abarcar ampliamente el rango de valores de policromismo, que viene

a ser de cero a cien ya que como hemos visto utilizamos un porcentaje de desviación para calcular el valor de policromismo. A continuación mostramos unos ejemplos de imágenes con las características que extrae el módulo de ellas.

2.3 Ejemplos de extracción de características de una imagen

Ejemplo 1:



POLICROMISMO 1: 4 colores de 12 pasan del 3% de presencia (se desvía un 25.0% de 2)
POLICROMISMO 2: 8 colores de 12 pasan del 0.5% de presencia (se desvía un 66.666664% de 4)
POLICROMISMO 3: 11 colores de 12 pasan del 0% de presencia (se desvía un 100.0% de 8)

VALOR FINAL DE POLICROMISMO: 63.888885%
COLOR DOMINANTE: Spring Green con el 40.3796%
CALIDOS: 68.158585% **FRIOS:** 31.841415%

Ejemplo 2:



POLICROMISMO 1: 6 colores de 12 pasan del 3% de presencia (se desvía un 50.0% de 2)
POLICROMISMO 2: 10 colores de 12 pasan del 0.5% de presencia (se desvía un 100.0% de 4)
POLICROMISMO 3: 11 colores de 12 pasan del 0% de presencia (se desvía un 100.0% de 8)

VALOR FINAL DE POLICROMISMO: 83.333336%
COLOR DOMINANTE: Ocean con el 51.695126%
CALIDOS: 8.945763% **FRIOS:** 91.05424%

Ejemplo 3:



POLICROMISMO 1: 2 colores de 12 pasan del 3% de presencia (se desvía un 0.0% de 2)
POLICROMISMO 2: 2 colores de 12 pasan del 0.5% de presencia (se desvía un 0.0% de 4)
POLICROMISMO 3: 3 colores de 12 pasan del 0% de presencia (se desvía un 0.0% de 8)

VALOR FINAL DE POLICROMISMO: 0.0%
COLOR DOMINANTE: Red con el 68.835464%
CALIDOS: 100.0% **FRIOS:** 0.0%

Ejemplo 4:



POLICROMISMO 1: 4 colores de 12 pasan del 3% de presencia (se desvía un 25.0% de 2)
POLICROMISMO 2: 5 colores de 12 pasan del 0.5% de presencia (se desvía un 16.666666% de 4)
POLICROMISMO 3: 8 colores de 12 pasan del 0% de presencia (se desvía un 0.0% de 8)

VALOR FINAL DE POLICROMISMO: 13.888888%
COLOR DOMINANTE: Ocean con el 35.257614%
CALIDOS: 56.862335% **FRIOS:** 43.13766%

Ejemplo 5:



POLICROMISMO 1: 6 colores de 12 pasan del 3% de presencia (se desvía un 50.0% de 2)

POLICROMISMO 2: 7 colores de 12 pasan del 0.5% de presencia (se desvía un 50.0% de 4)
POLICROMISMO 3: 8 colores de 12 pasan del 0% de presencia (se desvía un 0.0% de 8)

VALOR FINAL DE POLICROMISMO: 33.333332%
COLOR DOMINANTE: Spring Green con el 36.02247%
CALIDOS: 65.92082% FRIOS: 34.079178%

Ejemplo 6:



POLICROMISMO 1: 9 colores de 12 pasan del 3% de presencia (se desvía un 87.5% de 2)
POLICROMISMO 2: 10 colores de 12 pasan del 0.5% de presencia (se desvía un 100.0% de 4)
POLICROMISMO 3: 11 colores de 12 pasan del 0% de presencia (se desvía un 100.0% de 8)

VALOR FINAL DE POLICROMISMO: 95.833336%
COLOR DOMINANTE: Orange con el 23.323887%
CALIDOS: 59.088284% FRIOS: 40.911716%

Ejemplo 7:



POLICROMISMO 1: 5 colores de 12 pasan del 3% de presencia (se desvía un 37.5% de 2)
POLICROMISMO 2: 7 colores de 12 pasan del 0.5% de presencia (se desvía un 50.0% de 4)
POLICROMISMO 3: 10 colores de 12 pasan del 0% de presencia (se desvía un 66.666664% de 8)

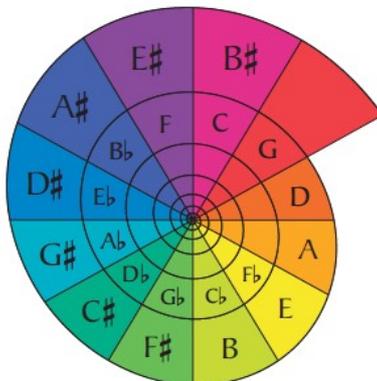
VALOR FINAL DE POLICROMISMO: 51.388885%
COLOR DOMINANTE: Ocean con el 47.938175%
CALIDOS: 11.62534% FRIOS: 88.374664%

3. Correspondencia entre colores y música

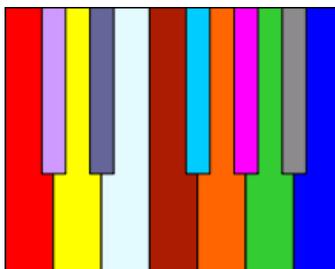
El compositor automático tiene tres parámetros de entrada claves a los que les tendremos que encontrar una correspondencia con los parámetros de salida del módulo anterior. Uno es la tonalidad, otro es el modo y el último es la velocidad (junto a este último se proporciona una velocidad máxima para poder situar el parámetro de velocidad, es decir, no es lo mismo velocidad 37 de 100 que 37 de 60). Este último parámetro hace referencia a la velocidad de la música como valor numérico, siendo el compositor el encargado de transformar ese parámetro en probabilidades de duración de nota y tempo concretas. Cada uno de los parámetros de entrada del compositor tendrá correspondencia con uno solo de los parámetros de salida del módulo de extracción de datos de colores.

3.1 Asociaciones

La primera asociación que haremos será la del color mayoritario (el color que tiene mas presencia en porcentaje en la imagen) con la tonalidad. La asociación concreta será la siguiente:



Aquí vemos la asociación entre los doce colores del círculo cromático y las doce notas musicales, pese a que en la imagen el tono de los colores parezca que no se corresponde exactamente. Los colores situados correlativamente son los mas cercanos el uno del otro en lo que a valor RGB se refiere. Estos colores correlativos siempre van a generar las composiciones de color mas armónicas posibles. De manera análoga, las notas situadas correlativamente también van a formar las composiciones más armónicas posibles dado que están situadas en intervalos de quintas, que es la distancia entre notas que suena mas armónica cuando esas notas suenan juntas. Se ha elegido esta asociación dentro de un variado abanico de posibilidades debido a que es la que mejor relaciona las notas musicales con los colores del círculo cromático, lo cual nos facilita la tarea al poder trabajar directamente con valores RGB concretos. Entre las diferentes alternativas que podíamos haber elegido tenemos la asociación de Newton entre colores y notas musicales que en vez de situar de manera correlativa las notas que mejor armonizan (las separadas por intervalos de quintas) sitúa correlativamente las que tienen la percepción sonora mas cercana (las notas correlativas en una escala musical). Por otro lado podíamos haber utilizado la asociación entre colores y notas de algún compositor musical sinestésico. La sinestesia es la habilidad de percibir mediante un sentido, estímulos normalmente perceptibles mediante otro sentido. En este caso tenemos por ejemplo al compositor Aleksandr Skriabin, que decía que podía oír colores mediante esta relación:



La segunda asociación que haremos será la del modo con el porcentaje de colores cálidos y fríos. Una imagen con predominancia de colores cálidos tendrá asociada una tonalidad mayor mientras que una imagen donde el porcentaje de colores fríos sea mayor que el de colores cálidos tendrá asociada una tonalidad menor. La razón de esto es que los colores cálidos dan sensación de actividad, alegría y dinamismo, algo que también transmiten las composiciones realizadas en tonalidades mayores. Sin embargo, una composición en un tono menor suele transmitir mas bien sentimientos que tienen que ver con la tranquilidad, la tristeza o la melancolía, algo que se asocia de la misma manera con los colores fríos.

La tercera asociación sera la del policromismo con la velocidad. Dado que el policromismo establecerá un porcentaje de desviación del número de colores relevantes esperado al número de colores relevantes encontrado asociaremos ese porcentaje a la velocidad, de manera que si tenemos un porcentaje de policromismo del 67% estableceremos la velocidad de composición a 67 de 100. Esta relación esta justificada en el hecho de que cuanto mayor sea el policromismo de una imagen mas caótica y agitada sera la sensación que transmite al visualizarla y por tanto la asociaremos a una mayor velocidad y agitación de la música. Por otro lado una imagen con apenas dos o tres colores relevantes producirá una sensación mas de calma y por eso generará una música mas lenta.

3.2 Tabla Resumen de parámetros

A modo de resumen y para clarificar los parámetros clave de cada módulo y especificar como están relacionados mostramos la siguiente tabla aclaratoria:

Parámetros de salida del módulo de extracción de datos	Color mayoritario (color con mas presencia en %)	Porcentaje de colores cálidos y fríos	Policromismo (porcentaje de desviación)
Parámetros de entrada del compositor musical asociados	Tonalidad asociada a ese color	Modo (mayor si el % de colores cálidos es mayor, si no menor)	Velocidad asociada a ese porcentaje de desviación

4. Módulo compositor musical automático

Como ya hemos comentado anteriormente, el compositor automático tiene tres parámetros clave que son la tonalidad, el modo y la velocidad (con respecto a una velocidad máxima). Hay que hacer notar que en realidad el compositor tiene mas parámetros de entrada como podrían ser la elección de acordes triadas o cuatriadas o la activación de la línea de bajo o no, pero son parámetros que no tienen correspondencia ni asociación con ninguna característica de la imagen y por tanto no resultan claves en el proyecto. Solo son parámetros extra con los que nos permite jugar el compositor. Los parámetros de tonalidad y modo son de transformación o aplicación directa. Los parámetros velocidad y máxima velocidad se procesan para generar un tempo y unas probabilidades de duración de nota concretas.

4.1 Cálculo de las probabilidades de duración de nota en base a la velocidad

Disponemos de ocho duraciones de nota diferentes, ordenadas de mayor duración a menor: redonda, blanca puntillo, blanca, negra puntillo, negra, corchea puntillo, corchea y semicorchea. La manera de establecer la probabilidad de que una nota tenga cada una de esas probabilidades es mediante un peso concreto que será relativo al peso total de todas las notas. De esta manera si por ejemplo tenemos unos pesos determinados por un vector de enteros como este {0,0,16,5,100,0,44,3} (cada posición del vector se asocia con una de las duraciones anteriores en el mismo orden que aparecen anteriormente), una corchea tendría $44/(16+5+100+44+3)$ probabilidades, o lo que es lo mismo, un 26% de ser la duración elegida para una nota.

En base a esto establecemos veinte velocidades con los pesos escritos a mano ordenadas de menor a mayor para cada una de las voces que tendrá la música generada (no tendremos las mismas probabilidades de duración de nota para la línea solista que para la línea de acompañamiento). Por ejemplo mostramos los pesos establecidos para cada una de las veinte velocidades de la línea solista:

```
private final int[] v0 = {6,4,55,0,40,0,0,0};
private final int[] v1 = {5,4,48,0,48,0,0,0};
private final int[] v2 = {4,4,42,0,55,0,5,0};
private final int[] v3 = {3,3,37,3,62,0,12,0};
private final int[] v4 = {2,3,33,4,69,0,18,0};
private final int[] v5 = {1,2,29,5,75,0,23,0};
private final int[] v6 = {0,2,25,6,82,0,28,0};
private final int[] v7 = {0,1,22,7,88,0,34,0};
private final int[] v8 = {0,0,19,8,94,0,39,0};
private final int[] v9 = {0,0,16,5,100,0,44,3};
private final int[] v10 = {0,0,14,5,90,1,48,6};
private final int[] v11 = {0,0,12,5,78,3,51,10};
private final int[] v12 = {0,0,8,5,66,3,60,16};
private final int[] v13 = {0,0,4,4,54,3,71,22};
private final int[] v14 = {0,0,2,4,42,4,85,31};
private final int[] v15 = {0,0,0,3,30,4,75,42};
private final int[] v16 = {0,0,0,3,18,5,65,53};
private final int[] v17 = {0,0,0,2,6,5,55,66};
private final int[] v18 = {0,0,0,1,6,6,45,79};
private final int[] v19 = {0,0,0,0,6,6,35,95};
private final int[] v20 = {0,0,0,0,1,6,25,112};
```

Vemos que las probabilidades de que la duración de nota sea una blanca en la velocidad mas lenta v0 serán de $55/105=52\%$ mientras que en la décima velocidad v9 ya solo serán de $16/168=9\%$.

Con todo esto lo único que nos falta por hacer es una simple regla de tres para transformar una velocidad de por ejemplo cincuenta y siete con un máximo de cien velocidades a su equivalente teniendo solo veinte velocidades, que en este caso sera la velocidad once ($20*57/100$) usando redondeo.

Si se han elegido veinte velocidades es porque creemos que proporcionan una variabilidad suficiente en la música generada como para que no sea necesario un mayor rango de velocidades. Pese a esto, una idea para mejorar este aspecto sería sacar en funciones el patrón de crecimiento/decrecimiento de los pesos de cada duración de nota conforme aumenta/disminuye la velocidad. De esta manera podríamos generar todo el rango de velocidades que quisiéramos siguiendo el mismo patrón que hemos establecido para estas veinte velocidades. Perderíamos algo de

precisión en los pesos ya que nunca podremos tener la precisión que tenemos al establecerlos a mano, pero ganaríamos en resolución (podríamos generar por ejemplo cien velocidades con el mismo patrón que estas veinte), y la pérdida de precisión en este caso no sería muy importante ya que partimos de que establecer unas probabilidades de duración de cada nota para una velocidad concreta es algo ya de por sí subjetivo.

A continuación mostraremos algunos ejemplos de las duraciones de nota que se acaban generando de manera aleatoria para los diferentes pesos que proporcionan las diferentes velocidades.

4.1.1 Ejemplos de notas generadas aleatoriamente usando los pesos de velocidades concretas

Velocidad 0 de 100:

EMin7 FMaj
CMaj7 EMin7

Velocidad 25 de 100:

GDom7 CMaj7
CMaj7 DMin7 AMin7

Velocidad 50 de 100:

AMin7 DMin7 GDom7
FMaj7 FMaj7

Velocidad 75 de 100:

CMaj7 DMin7 GDom7
CMaj7 DMin7 EMin7

Velocidad 100 de 100:

4.2 Cálculo del tiempo en base a la velocidad

Por otro lado también tendremos que generar un tiempo en base a la velocidad. La forma de calcular el tiempo es muy simple. Depende de tres parámetros: velocidad, máxima velocidad y rango. El rango es un rango de valores que le damos al tiempo para variar, estableciendo siempre el tiempo base a 100, de manera que si por ejemplo establecemos un rango de 14 el tiempo variará de 93 a 107 (un rango de 14). El rango es un valor estático que cuanto mas grande lo establezcamos mas variará el tiempo con respecto a la velocidad. Con una velocidad máxima de 100 y un rango de 14, a una velocidad de 0 el tiempo sería 93, a velocidad 50 sería 100 y a velocidad 100 sería 107. El resto de valores se distribuyen uniformemente con la lógica que se observa en el ejemplo.

4.3 Generación de música automática

Una vez que tenemos todos los parámetros que generan la música establecidos comenzaremos la generación de música de manera automática condicionada por estos parámetros. A groso modo, y centrándonos en los aspectos clave de la música que generaremos, tendremos una línea de acompañamiento base (progresiones de acordes) y una línea solista (melodía que sonará sobre los acordes).

4.3.1 Normas de composición básicas

La generación de música no será aleatoria. Las primeras reglas o condiciones que deberá tener la música generada por el compositor automático serán las que nos indican los parámetros de entrada, es decir, la tonalidad, el modo, el tempo y las probabilidades de duración de nota. Por tanto, teniendo en cuenta estas condiciones tendremos que la línea de acompañamiento se compondrá siempre de acordes que pertenezcan a la tonalidad y el modo establecidos, mientras que la línea solista contendrá siempre notas que se correspondan con la escala asociada a la tonalidad de la composición (si estamos componiendo en do mayor utilizaremos la escala de do mayor). Todos estos acordes y notas sonarán al tempo especificado y tendrán unas duraciones que se generarán con las probabilidades de duración de nota establecidas en los parámetros de entrada.

Estas son las primeras reglas o normas que seguirá el compositor automático de cara a generar la música, pero no son las únicas. De cara a intentar aumentar la calidad de la música compuesta se establecerán otras condiciones mas subjetivas y menos teóricas a seguir a la hora de generar la música.

4.3.2 Normas de composición adicionales para la progresión de acordes

Las primeras reglas adicionales que estableceremos serán las que afectan a la línea de acompañamiento o progresión de acordes. Los acordes generados no serán seleccionados aleatoriamente de entre todos los disponibles para la tonalidad establecida. En cambio, cada acorde tendrá unas probabilidades concretas de ser elegido, según cual sea el acorde que ha sonado antes que él. Por decirlo de otra manera, la progresión de acordes se generará teniendo en cuenta siempre unas probabilidades de transición de un acorde a otro concretas y establecidas mediante pesos. Estos pesos funcionan de la misma manera que lo hacen los pesos de duración de nota.

Los pesos de cada transición de un acorde a otro se han establecido a mano, pero están basados o mejor dicho, son una modificación subjetiva de un estudio estadístico sobre canciones reales (adjunto en la bibliografía que encontramos al final de este documento) donde se analiza un elevado número de canciones y se observa cual es el número de transiciones que hay de un acorde a otro para acabar estableciendo unas probabilidades reales de transiciones entre acordes. El problema de este estudio y del método que nosotros utilizamos en general es que se basa en información local y no global, es decir, que solo tiene en cuenta las transiciones de un acorde a otro de manera local sin tener en cuenta el contexto global en el que se producen esas transiciones. Básicamente solo tienen memoria sobre el acorde anterior. Esta es otra de las secciones que también se podrían mejorar realizando un estudio mas exhaustivo y creando otro método que tuviera en cuenta el contexto global en el que se produce una transición o que al menos tuviera mas memoria sobre acordes previos de cara a elegir el siguiente acorde a sonar. Pese a esto, con este método conseguimos condicionar al menos lo suficiente la progresión de acordes como para que no se produzcan transiciones que raramente se producen en la realidad y conseguimos que se produzcan mas a menudo transiciones como dominante → tónica que tanto vemos en la música estándar.

Los acordes disponibles para una tonalidad serán (abstrayéndonos de la tonalidad concreta usada) para los acordes triada del modo mayor el Imaj, IImaj, IIImin, IVmaj, Vmaj, VImin, VIIdim y para el modo menor el Imin, IIdim, IIImaj,

Ivmin, Vmin, Vlmaj, Vllmaj y los pesos de transición entre acordes serán los siguientes para el modo mayor:

```
int[] pesos1 = {8,17,10,17,32,17,0};
int[] pesos2 = {8,4,6,1,19,9,0};
int[] pesos3 = {1,2,1,7,1,7,0};
int[] pesos4 = {10,3,4,3,6,3,0};
int[] pesos5 = {10,3,3,4,2,6,0};
int[] pesos6 = {2,5,6,7,4,2,0};
int[] pesos7 = {17,0,3,0,1,0,0};
```

y los siguientes para el modo menor:

```
int[] pesos1 = {9,10,10,24,40,10,19};
int[] pesos2 = {9,2,6,1,13,1,6};
int[] pesos3 = {1,2,1,7,5,3,3};
int[] pesos4 = {13,6,4,3,8,2,4};
int[] pesos5 = {11,0,2,4,3,6,4};
int[] pesos6 = {4,0,1,3,3,1,3};
int[] pesos7 = {8,0,2,1,3,3,1};
```

4.3.3 Normas de composición adicionales para la melodía

Como ocurre con los acordes, las notas de la melodía tampoco se elegirán de manera aleatoria de entre todas las disponibles en la escala que estemos utilizando. En este caso en vez de intentar condicionar la elección de las notas basándonos en un análisis estadístico sobre transiciones entre notas en melodías reales lo que haremos será intentar mediante la observación de melodías reales establecer ciertas normas que condicionen la progresión de notas de la melodía. La razón de no intentar basar las reglas de transición entre notas en un estudio estadístico es que si ya en las progresiones de acordes se acababa extrayendo información local que carecía de contexto global, en las progresiones de notas esto se acentúa incluso más ya que van a haber muchas más notas que acordes y las transiciones van a ser mucho más rápidas, lo que hace que sea aun más importante el contexto en el que se ha producido esa transición de una nota a otra (por ejemplo si una melodía está simplemente arpegiando las notas del acorde de do mayor, la transición de notas será do mi sol y nosotros nos quedaremos con que después de mi se ha transicionado a sol pero sin tener en cuenta que es porque está arpegiando un acorde y daremos valor al hecho de que estando en mi transicionamos a sol).

Lo que haremos en este caso será establecer un peso inicial de transición entre las notas de la escala igual para todas las notas, lo que quiere decir que en principio desde cualquier nota de la escala se tendrá las mismas posibilidades de transicionar a cualquiera de las otras notas. Luego, y en función de ciertos aspectos, se aumentará o disminuirá el peso para transicionar a una cierta nota concreta. En concreto, aumentaremos el peso de las notas que se corresponden con las notas que forman el acorde que esta sonando en ese momento (si esta sonando do mayor las notas do mi sol aumentarán su peso y por lo tanto sus probabilidades de ser elegidas en la progresión aumentaran también). De la misma manera, también aumentaremos el peso de las notas que son adyacentes a la última nota tocada (si la última nota añadida a la progresión fue un re, tanto do como mi aumentarán también su peso). Otro factor que tendremos en cuenta para aumentar el peso de una nota es que esté en la dirección (ascendente o descendente) de la progresión de notas, es decir, si la progresión viene siendo do re mi aumentaremos el peso de fa y de las notas que lo siguen, lo que hará que haya más probabilidades de seguir con una progresión de notas ascendente que de volver a re o a do cortando la progresión ascendente de la melodía. Por otro lado disminuirémos el peso de la nota que acaba de sonar (lo cual hará que haya menos probabilidades de repetir nota que de tocar otra nota cualquiera). En principio todos los pesos iniciales están establecidos a dos y cada condición de aumento de peso aumenta en otra unidad el peso o lo disminuye si es una condición de disminución de peso.

Todas estas condiciones que aumentan o disminuyen peso lo único que hacen es eso, aumentar o disminuir el peso con respecto a las demás notas, o lo que es lo mismo, aumentar o disminuir sus probabilidades de ser elegidas, pero en ningún caso restringen posibilidades para no eliminar totalmente el carácter aleatorio de la generación de música.

5. Dificultades encontradas en el proyecto

Algunas de las dificultades encontradas como el ajuste de los parámetros utilizados para medir el policromismo ya han sido mencionadas y discutidas. Hay otras, sin embargo, que intentaremos explicar a continuación.

La mayor dificultad del proyecto reside en lo que concierne a la generación automática de música. Conseguir generar música mediante un algoritmo es relativamente fácil, pero conseguir que este algoritmo genere buena música ya es más difícil, sobre todo teniendo en cuenta lo ya de por sí subjetivo de este término.

Es difícil además, debido a que cuanto más intentas condicionar la generación de música estableciendo reglas para que suene mejor, más la encorsetas. Dicho de otra manera, el problema surge debido a que la idea es que se genere música de manera continua y aleatoria, intentando que esta suene de manera coherente, pero al mismo tiempo para conseguir que una melodía sea coherente y bonita tienes que aplicar unas reglas que reducen la libertad de esa composición aleatoria, pudiendo llegar a estar demasiado condicionada.

Esto se suma al hecho ya mencionado de lo subjetivo que es considerar una melodía como buena o bonita. No hay unas normas concretas que establezcan lo que va a sonar mejor o peor, o lo que va a tener más elegancia o menos, las únicas normas establecidas son las que proporciona la teoría musical, pero solo con estas no vas a conseguir hacer buena música, o si no no habría compositores profesionales. E incluso resulta difícil pensar que un compositor

profesional tenga métodos para crear buenas melodías que se puedan transformar en normas y se puedan aplicar con algoritmos.

Todo esto en realidad era de prever dado que la música tiene que ver con la imaginación, la creatividad y el arte, todas ellas cosas difíciles de combinar con la informática o la automatización. Igualmente se ha hecho un esfuerzo por resolver estos problemas de la mejor manera posible, intentando por un lado aplicar todos los conocimientos musicales teóricos y también los subjetivos propios que fueran posibles de automatizar y convertir en reglas para la generación de música y por otro lado dar a la música la mayor libertad posible estableciendo las mínimas reglas necesarias para proporcionar una música coherente y decente y que al mismo tiempo conservara el suficiente grado de aleatoriedad.

6. Manual de usuario (interfaz)

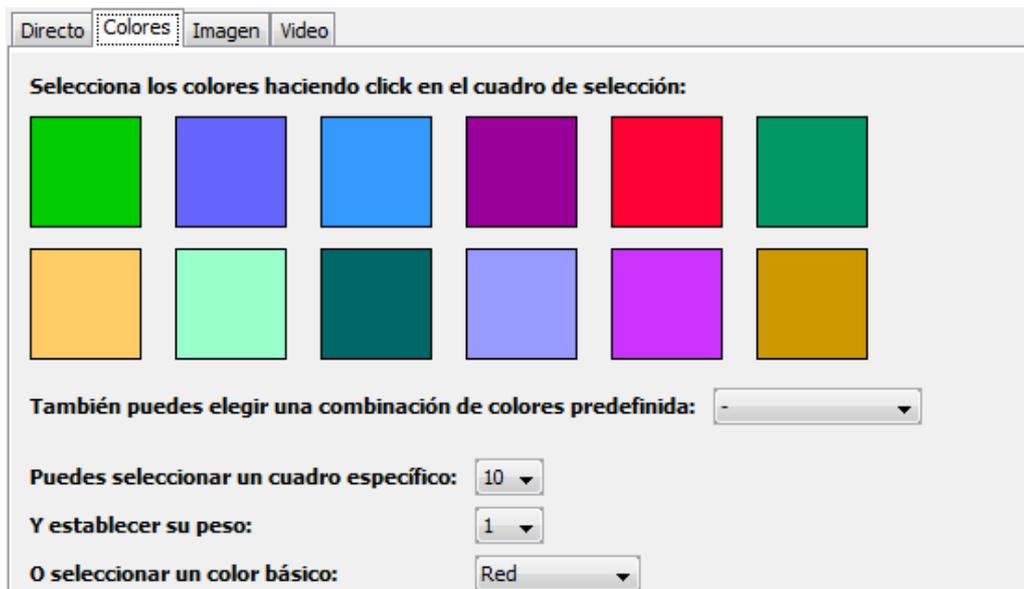
A lo largo de esta memoria se han mencionado los aspectos clave del proyecto, dejando de lado otras funcionalidades y características menos relevantes pero que al fin y al cabo han sido implementadas. Por ello y como ayuda de cara a la utilización de la interfaz de la aplicación que acopla los módulos explicados anteriormente se muestra este breve manual de usuario.

The screenshot displays the application's settings interface. At the top, there are tabs for 'Directo', 'Colores', 'Imagen', and 'Video'. The main settings area is divided into four columns: 'Elige un instrumento:', 'Elige el modo:', 'Elige la tonalidad:', and 'Tipo de acordes:'. Under 'Elige un instrumento:', 'Piano / Cello' is selected. Under 'Elige el modo:', 'Mayor' is selected. Under 'Elige la tonalidad:', 'C' is selected. Under 'Tipo de acordes:', 'Cuatriadas' is selected. Below these are a velocity slider and two checkboxes: 'Activar línea de bajo:' (set to 'No') and 'Activar línea de armonización:' (set to 'No'). A summary bar shows 'Color dominante: -', 'Cálidos: -', 'Fríos: -', 'Policromismo: -', 'Tonalidad: Do', 'Modo: Mayor', and 'Velocidad: 50/100'. At the bottom, a musical score is shown with two staves. The first staff has chords CMaj7, EMin7, CMaj7, and GDom7. The second staff has chords EMin7 and CMaj7.

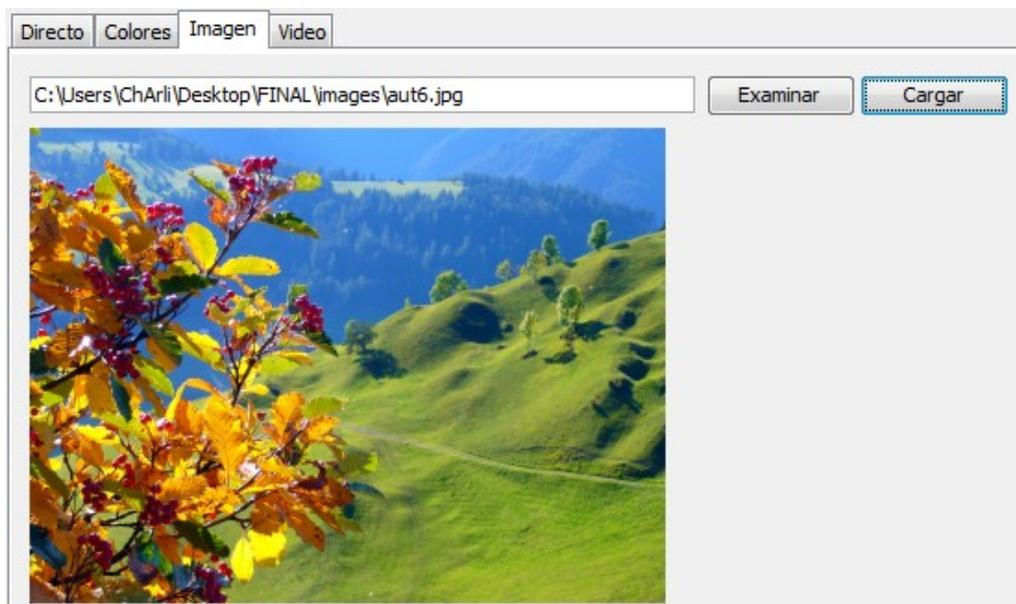
En esta primera ventana tenemos la opción de generar música estableciendo los parámetros del compositor de manera directa, sin extraer datos de colores y establecer una correspondencia con los datos musicales. Las opciones extra que nos permite esta opción son cambiar los instrumentos que por defecto son el piano y el chelo a la guitarra y el bajo, cambiar los acordes de triada a cuatriada, activar la armonización, lo cual hará que suene una melodía aparte de la melodía solista que consistirá en la replicación de la misma melodía a una distancia de una tercera y por último permite activar la línea de bajo que consistirá en una línea de bajo que improvisará sobre las notas del acorde que este sonando.

En el cuadro de en medio tenemos los parámetros que hemos elegido (en este caso) o los parámetros que se han generado en base a la paleta de colores o la imagen (en los siguientes casos).

En la parte de abajo tenemos una representación en partitura de la música generada que esta sonando.



En esta segunda ventana tenemos la opción de modificar a nuestro antojo una paleta de colores que en el fondo es como si fuera una imagen, siendo cada uno de los cuadros a los que podemos dar un color, equivalente a lo que un píxel en una imagen es. Podemos modificar el color de cada uno de los cuadros haciendo clic en él y eligiendo un valor RGB cualquiera, o podemos seleccionarlo en el menú de abajo y elegir un color concreto de los doce del círculo cromático. También podemos establecer pesos diferentes para cada uno de los cuadros, que lo que hace en el fondo es replicar ese cuadro, es decir, si tenemos un cuadro con peso tres lo que ocurre es que el resultado es equivalente al que tendríamos si tuviéramos dos cuadros mas aparte de todos los ya presentes, con el mismo valor que el del cuadro que esta siendo replicado. Por otro lado podemos establecer una combinación global de colores ya predefinida con combinaciones de colores cálidos,fríos,etc.



En esta última ventana simplemente cargamos una imagen y el programa extrae los datos sobre los colores que tiene. Luego obtiene los correspondientes parámetros del compositor musical asociados a esos datos sobre colores y genera música en base a esos parámetros.

Faltaría por desarrollar la cuarta ventana. La idea de la cuarta ventana es bastante simple y se trata de realizar lo mismo que realizamos en la tercera ventana con una imagen, pero con un video. Dado que un video no es mas que una sucesión de frames (imágenes) y debido a que el análisis de una imagen requiere un cierto tiempo, habría que analizar todos los frames del video que diera tiempo, es decir, siempre que acabara de analizar un frame pasaría a generar música en base a los parámetros que ha generado ese frame y comenzaría ya a analizar el siguiente.

7. Bibliografía y enlaces de interés

Enlace al estudio estadístico de las transiciones entre acordes
<http://theory.esm.rochester.edu/temperley/kp-stats/index.html>

Enlace a la explicación detallada de la relación entre nuestra paleta de colores y las tonalidades
<http://www.virtuosoism.com/Color-Sound%20Relationships.pdf>

Enlace sobre teoría de la armonía y psicología de los colores
<http://s3.accesoperu.com/wp6/includes/htmlarea/mezclador/ayuda/ec.htm>

Enlace sobre el compositor Aleksandr Scriabin y su habilidad sinestésica
http://es.wikipedia.org/wiki/Alexander_Scriabin