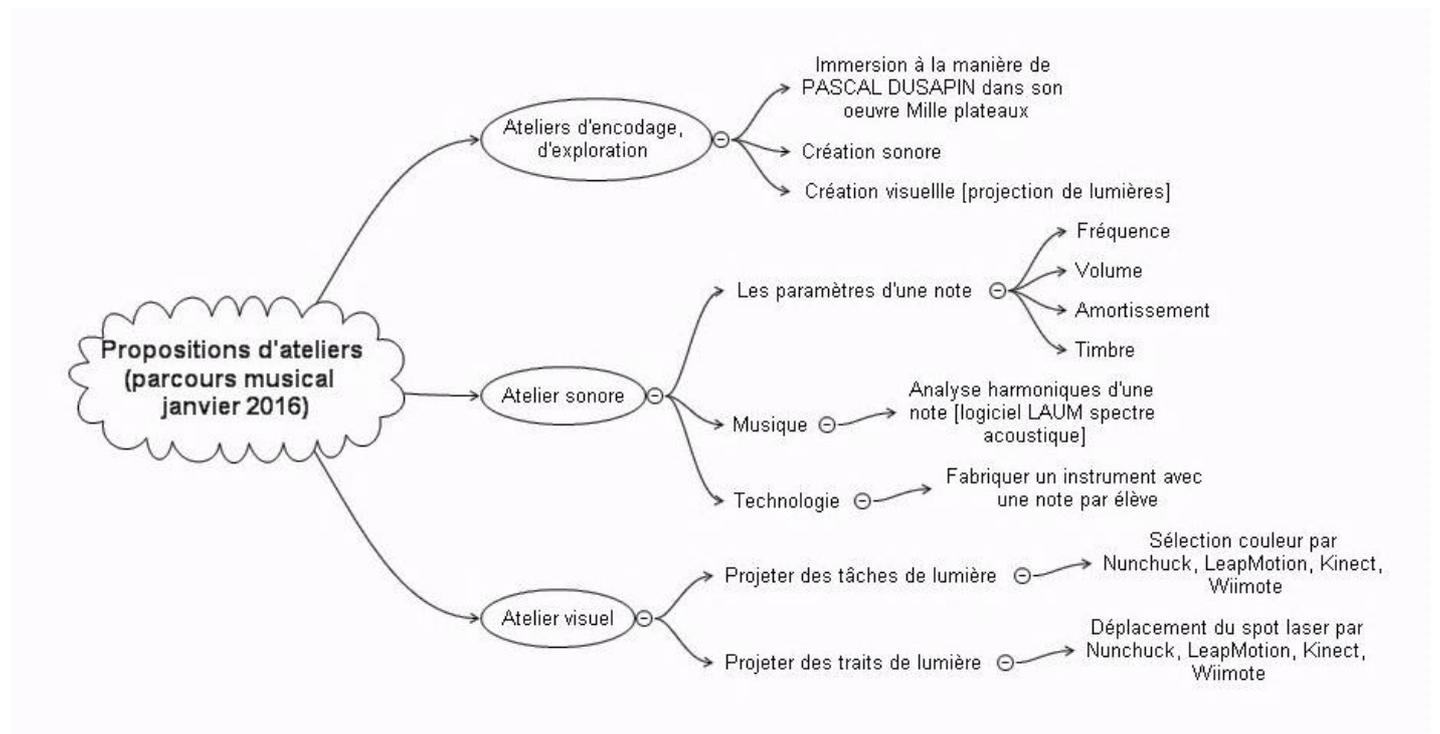


Ateliers sonores et visuels

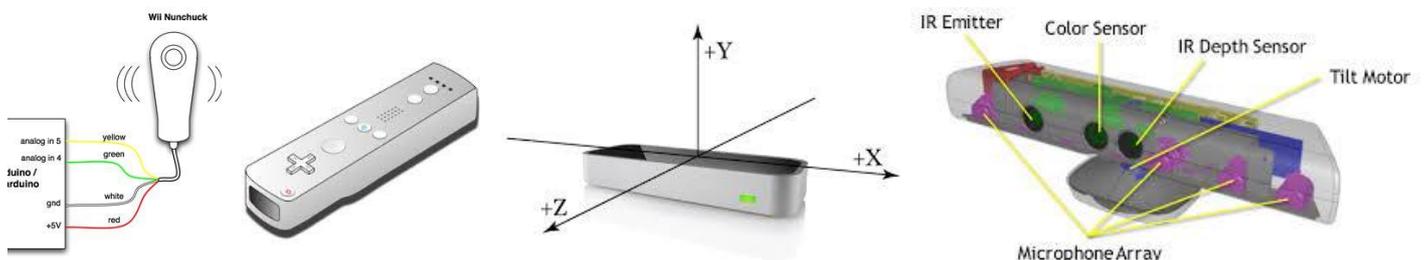
Création en amont de la restitution du 2 mai

Parcours en lien avec l'œuvre de PASCAL DUSAPIN



Ateliers de création les 26, 27 et 28 avril 2016

Restitution lundi 2 mai 2016 Lycée Livet



Fiches proposées pour l'accompagnement de la création sonore

RECHERCHE DE LA STRUCTURE DU MORCEAU SON/LUMIERE [fiches présentes en annexe]

FICHE n° 1 - Autour de la forme

* Inciter rapidement le groupe à prendre conscience de la structure générale souhaitée pour le morceau avant toute construction.

* Les mélodies créées auront une couleur particulière en fonction de leur partie. (par exemple, une mélodie peu rythmée pour la partie couplet et au contraire très tonique dans la partie refrain)

Plusieurs choix possibles pour le groupe

* ABA

* ABA'

* forme couplet/refrain

> s'il reste du temps, possibilité d'insérer une introduction et une conclusion à la création

CREATION MUSICALE [fiches présentes en annexe]

Recherche de mélodies autour du piano (gommettes positionnées sur les touches de chaque note de la gamme)

* ne pas hésiter à laisser les élèves tâtonner sur le piano

* les inciter à transcrire les correspondances de fréquences pour chaque note de la gamme (> leur préciser qu'une note identique peut être jouée dans différentes octaves)

4 choix

* FICHE n°2 - Gamme **pentatonique** Do mineur

* FICHE n°3 - Gamme **pentatonique** Ré mineur

* FICHE n°4 - Gamme **pentatonique** Mi Majeur

* FICHE n°5 - Mélodie libre (sans contrainte de gamme)

> cette fiche n°5 se destine à des élèves qui pourraient être musiciens et qui auraient l'envie de créer des mélodies simples sans aide des gommettes et sans gamme prédéfinie à l'avance

Après avoir découvert les notes de la gamme (ou bien choisie leur notes pour le groupe de la FICHE n°5) :

* FICHE n°6 - Création Mélodie ELEVE

> Les élèves dans un premier temps associent 2 ou 3 sons ensemble, tâtonnent, pratiquent autour du piano pour se mettre d'accord sur 3 ou 4 mélodies. Ils associent également à cette mélodie une rythmique propre

> dans un second temps, une fois le choix définitif, ils peuvent transcrire les différentes associations/succession de notes sur le FICHE n°6

> ils peuvent également s'enregistrer à l'aide de leur téléphone afin de pouvoir reproduire la même rythmique une fois les notes codées et les touches/interrupteurs connectés

CODAGE GRAPHIQUE [fiches présentes en annexe]

Dernière étape de la création > IMPORTANTE et pouvant PRENDRE du TEMPS

* FICHE n°7 - Exemples de partitions graphiques + idées codage

> exemples de différentes partitions graphiques (avec vidéos afin d'associer le son à la lecture de la partition VIDEO 1 > Cathy BERBERIAN et/ou VIDEO 2 > Artikulation)

> propositions de codages possibles en fonction des différents paramètres musicaux

* FICHE n° 8 - Partition codage CREATION - Fichier vierge

> Fichier vierge pouvant servir de base à la construction graphique de la création

> 7 ou 8 fichiers nécessaires par groupe

Atelier SON : définition des paramètres d'une note

Activité de codage de notes avec différents paramètres à l'aide du sketch Processing.
Il s'agit de modifier les codes sonores associés aux touches du clavier de l'ordinateur [azerty...][qsdrg...][wxcvb...]



Fiche activité :

Une note est définie avec les paramètres :

- Volume
- Fréquence
- Timbre
- Amortissement, Durée.

Les élèves réalisent le codage informatique de notes

Activité :

Ouvrir le logiciel Processing et charger le programme "ProjetSon.pde".

Sélectionner l'onglet "clavier"

Proposer aux élèves de modifier les paramètres comme ci-dessous :

Lancer le programme par l'icône en forme de triangle [haut à gauche de la barre du menu]

Modifier les paramètres

Sélectionner l'onglet 'clavier'

```
void keyPressed()
{
  SineWave mySine;
  TriangleWave myTriangle;
  SquareWave mySquare;
  MyNoteSin newNoteSin; // Sinus
  MyNoteTriangle newNoteTriangle; // triangle
  MyNoteSquare newNoteSquare; // carré

  //carte des touches en Hz (note)
  float note = 0;
  switch(key) // note => Timbre, fréquence, durée amortissement EX : MyNoteSin 0.8=volume, 0.55=amortissement
  {
    /* */
    case 'a': {newNoteSin = new MyNoteSin(260, 0.8, 0.95);} break;
    case 'z': {newNoteSin = new MyNoteSin(300, 0.6, 0.55);} break;
    case 'e': {newNoteSin = new MyNoteSin(340, 0.6, 0.6);} break;
    case 'r': {newNoteSin = new MyNoteSin(380, 0.6, 0.65);} break;
    case 't': {newNoteSin = new MyNoteSin(420, 0.6, 0.7);} break;
    case 'y': {newNoteSin = new MyNoteSin(460, 0.6, 0.75);} break;
    case 'u': {newNoteSin = new MyNoteSin(500, 0.6, 0.8);} break;
    case 'i': {newNoteSin = new MyNoteSin(540, 0.6, 0.85);} break;
    case 'o': {newNoteSin = new MyNoteSin(580, 0.6, 0.9);} break;
    case 'p': {newNoteSin = new MyNoteSin(620, 0.6, 0.95);} break;
    case 'q': {newNoteTriangle = new MyNoteTriangle(261, 0.3, 0.9);} break;
    case 's': {newNoteTriangle = new MyNoteTriangle(277, 0.6, 0.55);} break;
    case 'd': {newNoteTriangle = new MyNoteTriangle(293, 0.6, 0.6);} break;
    case 'f': {newNoteTriangle = new MyNoteTriangle(311, 0.6, 0.65);} break;
    case 'g': {newNoteTriangle = new MyNoteTriangle(329, 0.4, 0.7);} break;
    case 'h': {newNoteTriangle = new MyNoteTriangle(349, 0.6, 0.75);} break;
    case 'j': {newNoteTriangle = new MyNoteTriangle(369, 0.6, 0.8);} break;
    case 'k': {newNoteTriangle = new MyNoteTriangle(392, 0.8, 0.85);} break;
  }
}
```

Chaque touche est définie par une fréquence, une amplitude, un amorti et un timbre.
Exemples de ligne de code associée à la touche [a] :

Pour une note d'un timbre "sinusoïdal" associée à la touche du clavier "a"
 Pour une fréquence de 260Hz
 Pour un volume sonore de 60% [0.6]
 Pour un amorti [d'une note impulsionnelle jusqu'à maintenue de 0 à 0.99]
 La durée sera liée au rythme d'appui sur les touches du clavier...

```
case 'a': {newNoteSin = new MyNoteSin(260, 0.6, 0.95);} break;
```

De la même manière

Pour une note d'un timbre "triangulaire" associée à la touche du clavier "q"

```
case 'q': {newNoteTriangle = new MyNoteTriangle(440, 0.6, 0.9);} break;
```

Pour une note d'un timbre "rectangulaire" associée à la touche du clavier "w"

```
case 'w': {newNoteSquare = new MyNoteSquare(174, 0.6, 0.9);} break;
```

Associer à chaque touche du clavier la gamme chromatique de do à do4 (do, do#, ré, ré#...)

- Les fréquences sur présentées ci-dessous dans le cadre rouge
- Choisir le timbre souhaité (sinusoïdal, triangulaire ou carré)

Tester le codage des touches ainsi programmées.

Analyser le spectrogramme lors de la génération de chaque note.

Fréquences des notes (en hertz) dans la gamme tempérée

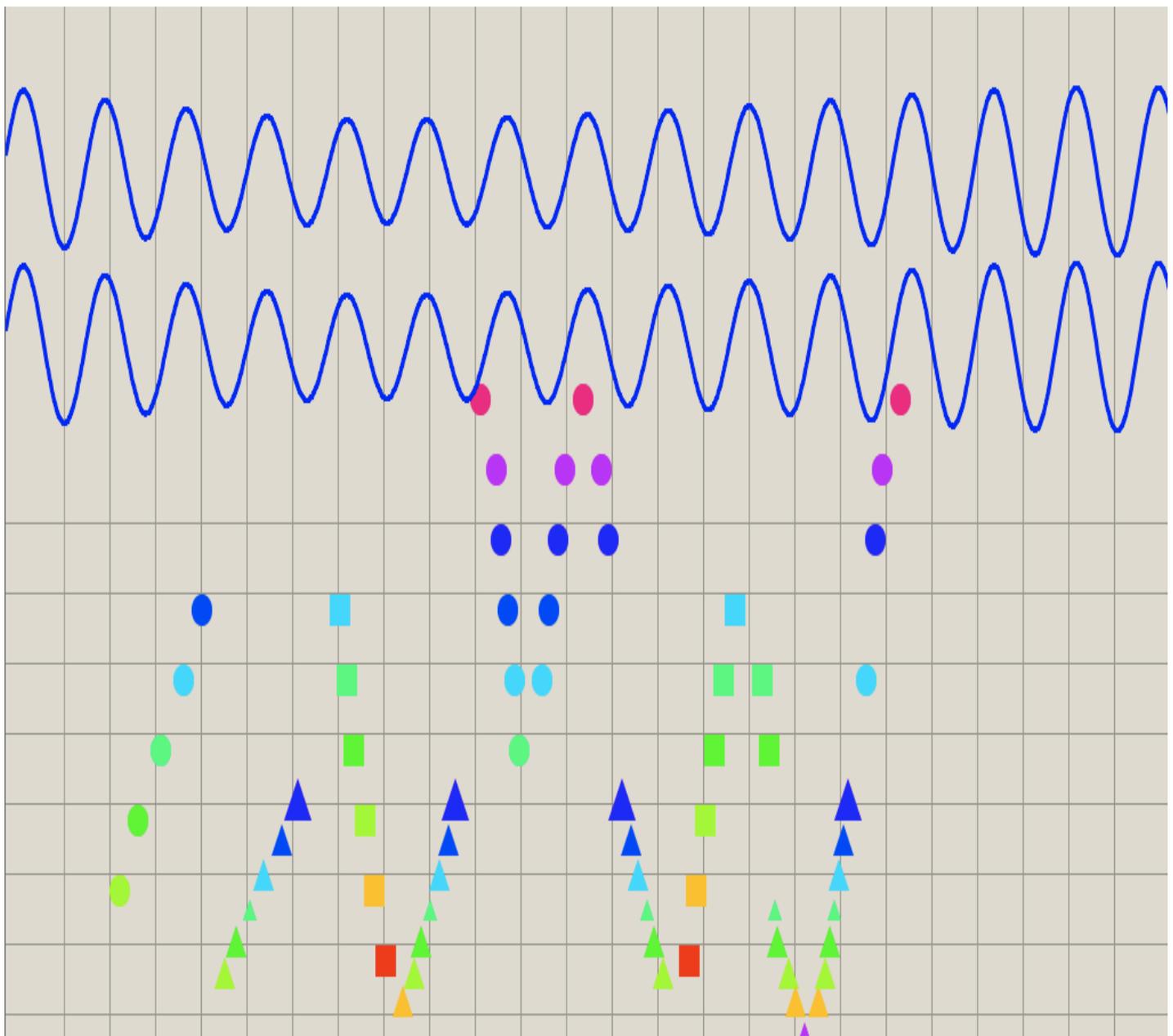
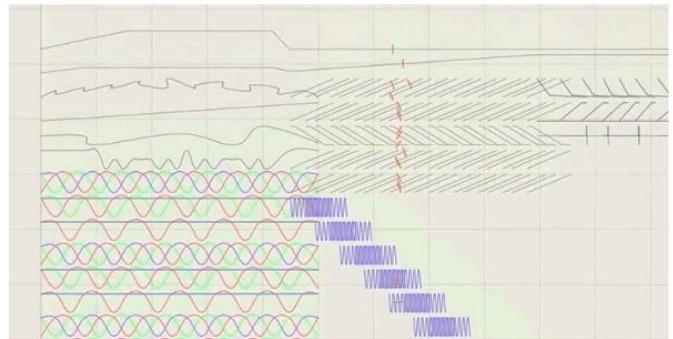
Note\octave	0	1	2	3	4	5	6	7
do ou si#	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
do# ou réb	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92
ré	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64
ré# ou mi b	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03
mi ou fa b	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04
fa ou mi#	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
fa# ou sol b	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
sol	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
sol# ou lab	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
la	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00
la# ou sib	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62
si ou do b	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13

Chaque élève choisit une note avec ses paramètres
 [Volume, Fréquence, Timbre, Amortissement-durée]

Chaque élève paramètre "sa" note.

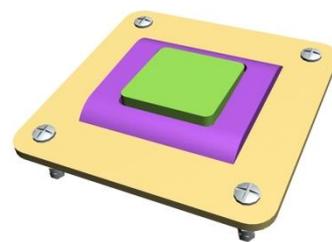
Faire si nécessaire un tableau des paramètres des notes de chaque élève.

Le Programme Processing construit en temps réel un graphique image des notes jouées dans l'esprit du logiciel Iannix utilisé par l'équipe de Pascal Dusapin, Thierry Coduys et Guillaume Jacquemin :



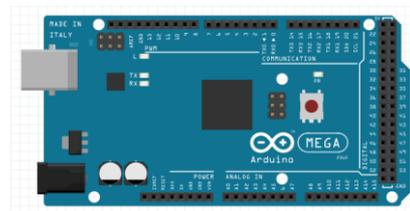
Instrument collectif [Connexion Carte Arduino]

L'idée est de proposer une deuxième partie d'activité où les notes ne seront plus générées seulement par le clavier mais également par des contacts électriques reliés à une carte interface.



Besoin technique :

Le matériel précédent, une carte interface, des contacts électriques [un contact par élève].



Fiche activité :

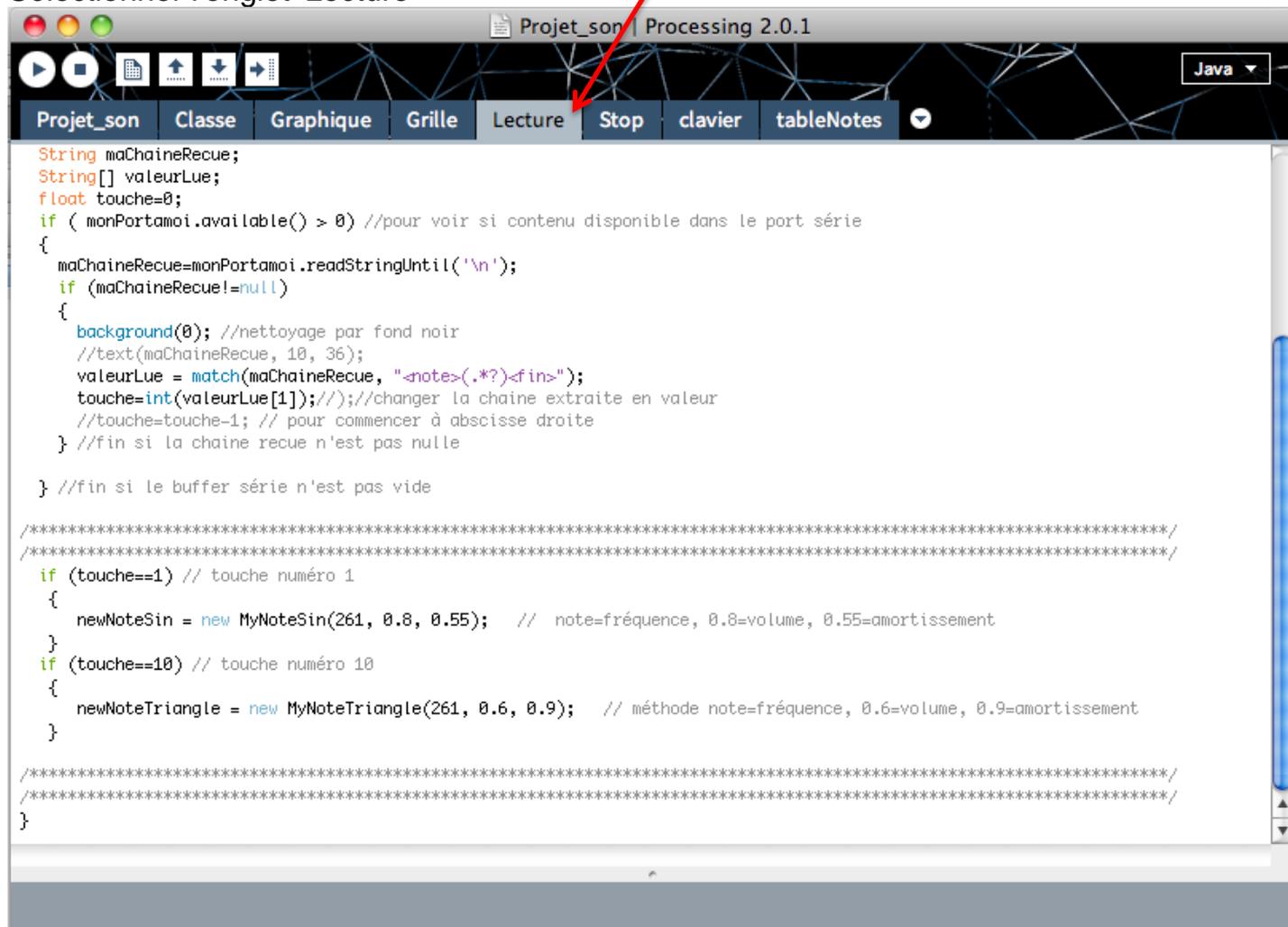
Interagir pour créer un instrument collectif de synthèse de notes. Chaque contact électrique est paramétré par une ligne de code comme précédemment.

Activité :

Paramétrer les notes par chaque élève.

Dans le logiciel Processing :

Sélectionner l'onglet 'Lecture'



```
String maChaineRecue;
String[] valeurLue;
float touche=0;
if ( monPortamoi.available() > 0) //pour voir si contenu disponible dans le port série
{
  maChaineRecue=monPortamoi.readStringUntil('\n');
  if (maChaineRecue!=null)
  {
    background(0); //nettoyage par fond noir
    //text(maChaineRecue, 10, 36);
    valeurLue = match(maChaineRecue, "<note>(.*?)<fin>");
    touche=int(valeurLue[1]);//;//changer la chaine extraite en valeur
    //touche=touche-1; // pour commencer à abscisse droite
  } //fin si la chaine recue n'est pas nulle
} //fin si le buffer série n'est pas vide

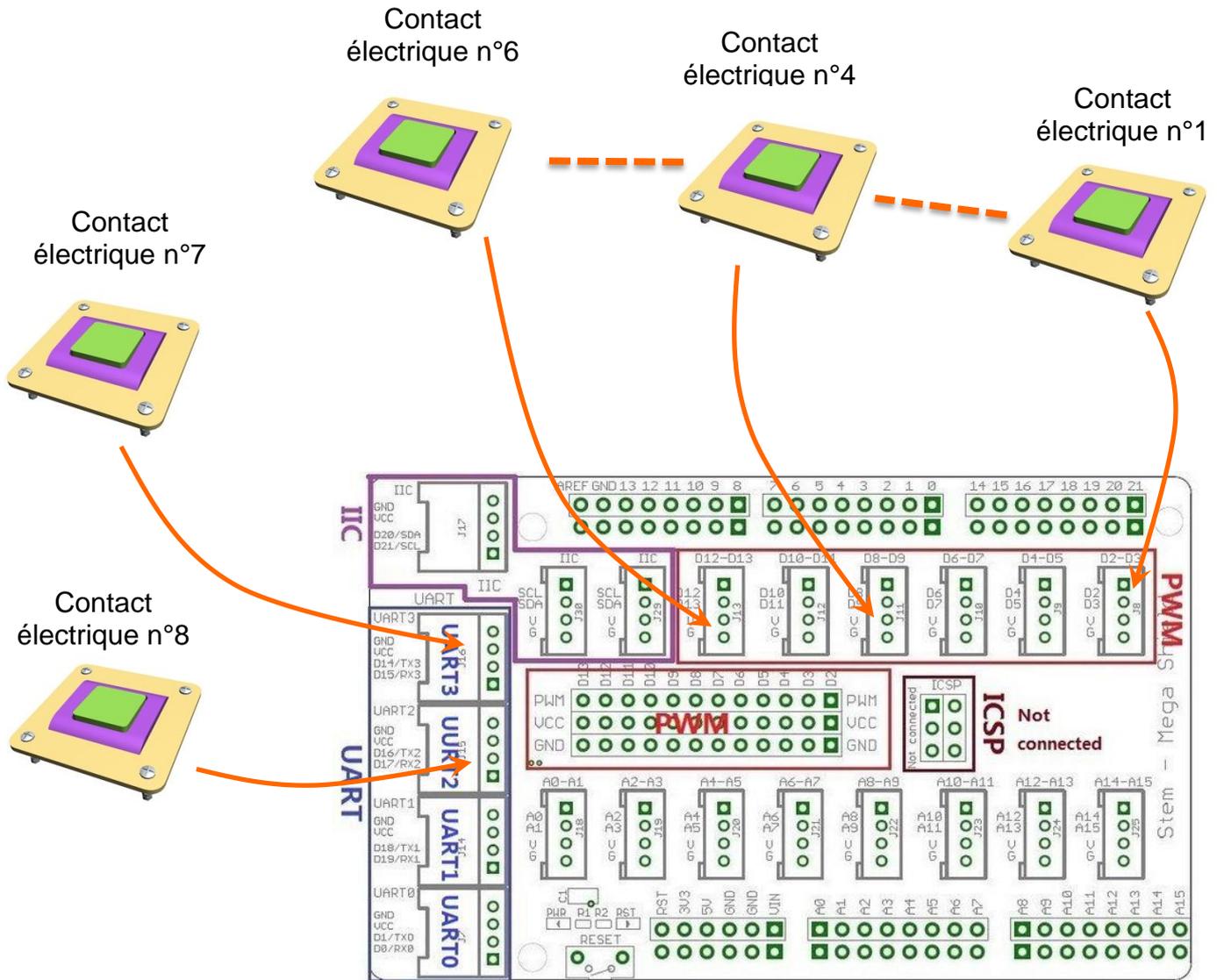
/*****
/*****
if (touche==1) // touche numéro 1
{
  newNoteSin = new MyNoteSin(261, 0.8, 0.55); // note=fréquence, 0.8=volume, 0.55=amortissement
}
if (touche==10) // touche numéro 10
{
  newNoteTriangle = new MyNoteTriangle(261, 0.6, 0.9); // méthode note=fréquence, 0.6=volume, 0.9=amortissement
}

/*****
/*****
}
```

Programmer des notes qui seront associées à des touches [contacts électriques numérotés de 1 à 8].

Connexion des 8 contacts électriques à la carte Arduino :

Raccorder chaque contact entre la carte Arduino et ses bornes via un cordon comme suit :



Tester votre câblage soit directement sous le logiciel Processing soit en lançant le code "ArduinoNotesContacts Elec.ino" sous le logiciel Arduino :

```
ArduinoNotesContactsElec | Arduino 1.6.5
Fichier Edition Croquis Outils Aide
ArduinoNotesContactsElec
int note=0;
boolean contact[53];

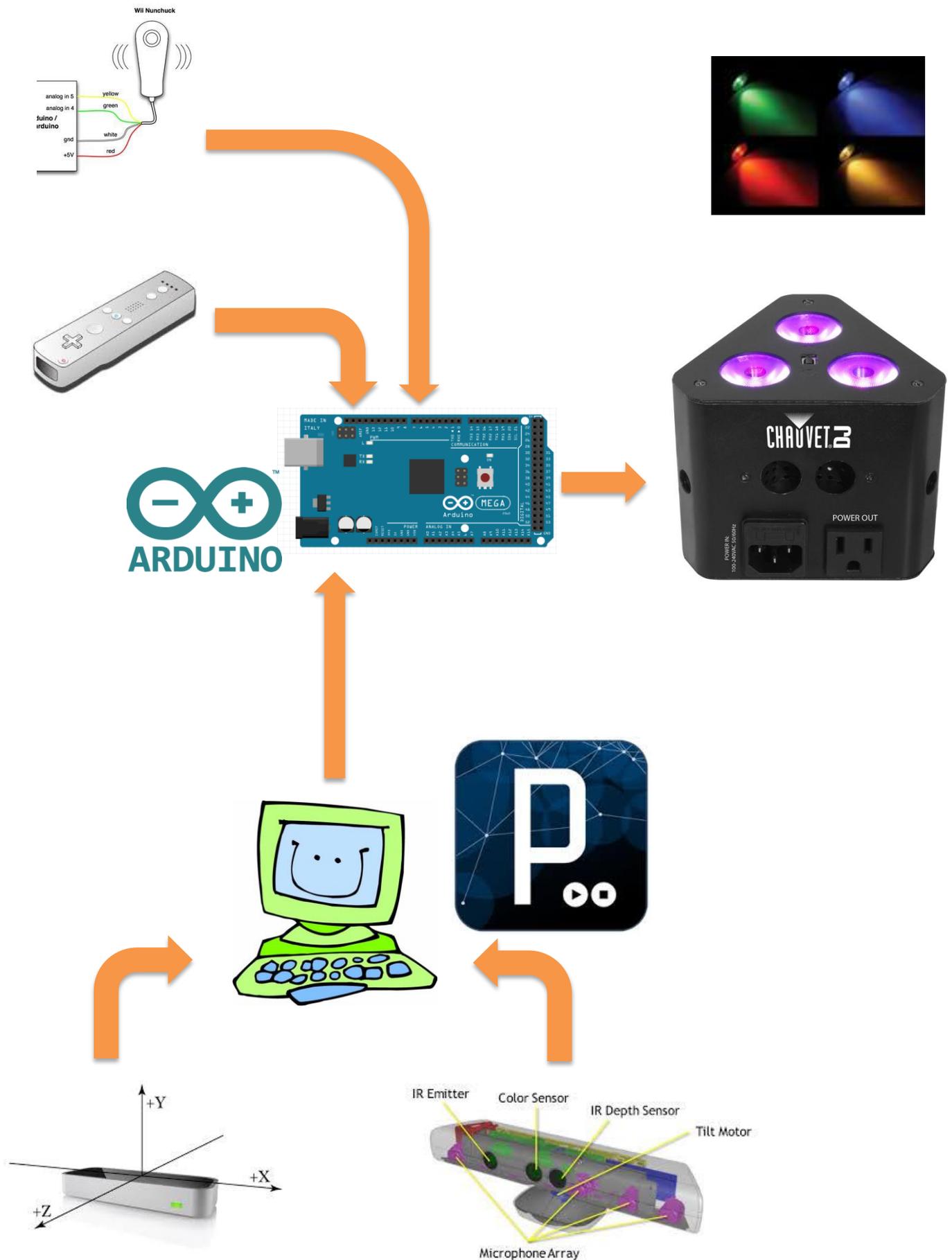
void setup()
{
  Serial.begin(38400);
  for (int i=2; i<=19; i++)
  {
    pinMode(i, INPUT);
    digitalWrite(i, HIGH); // R pull-up
  }
  pinMode(13, OUTPUT);
  delay(200);
}

void EnvoiMessage(int numeroContact)
{
  Serial.print("<note>");
  Serial.print(numeroContact);
}

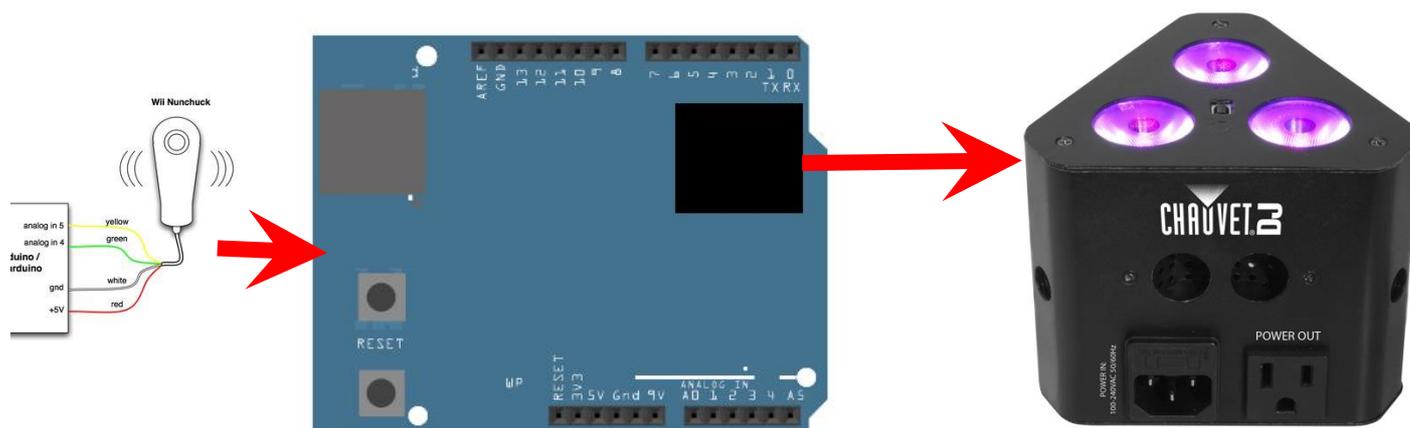
Téléversement terminé
```

```
COM11 (Arduino Mega or Mega 2560)
<note>1<fin>
<note>2<fin>
<note>3<fin>
<note>4<fin>
<note>5<fin>
<note>6<fin>
<note>7<fin>
<note>8<fin>
<note>9<fin>
Défilement automatique Pas de fin de ligne 38400 baud
```


Atelier Lumière 1 : Projeter des tâches de lumière



Commande SPOT DMX via manette Nunchuck



Un spot scénique dispose d'une connexion numérique [DMX] qui permet de régler le niveau de rouge, de vert et de bleu présents dans la lumière émise. L'envoi de commandes numériques permet de moduler la lumière produite par un spot.

Besoin technique :

Un PC

Le logiciel de développement Arduino

Une carte électronique programmable Arduino Nano + connecteur Nunchuck

Une manette de jeu Nunchuck.

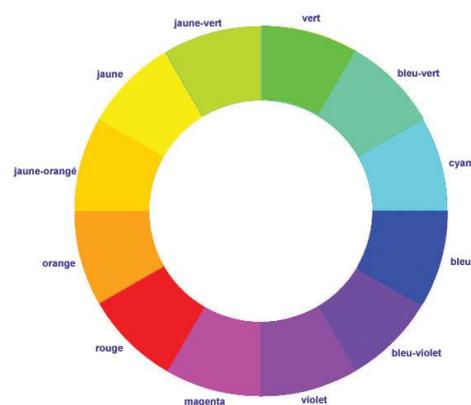
Un spot scénique numérique [à commande DMX]

Fiche activité :

Une couleur quelconque peut être constituée d'une addition de couleurs primaires (Rouge, Vert, Bleu)

Un spot scénique peut moduler chaque composante primaire de couleur.

Les élèves réalisent le codage de couleur en paramétrant les niveaux Rouge, Vert, Bleu émis.



Activité :

Câbler la carte électronique [cette partie peut être shuntée]

Ouvrir le logiciel Arduino et lancer le programme " ArduinoCommandeDMX_USB.ino "

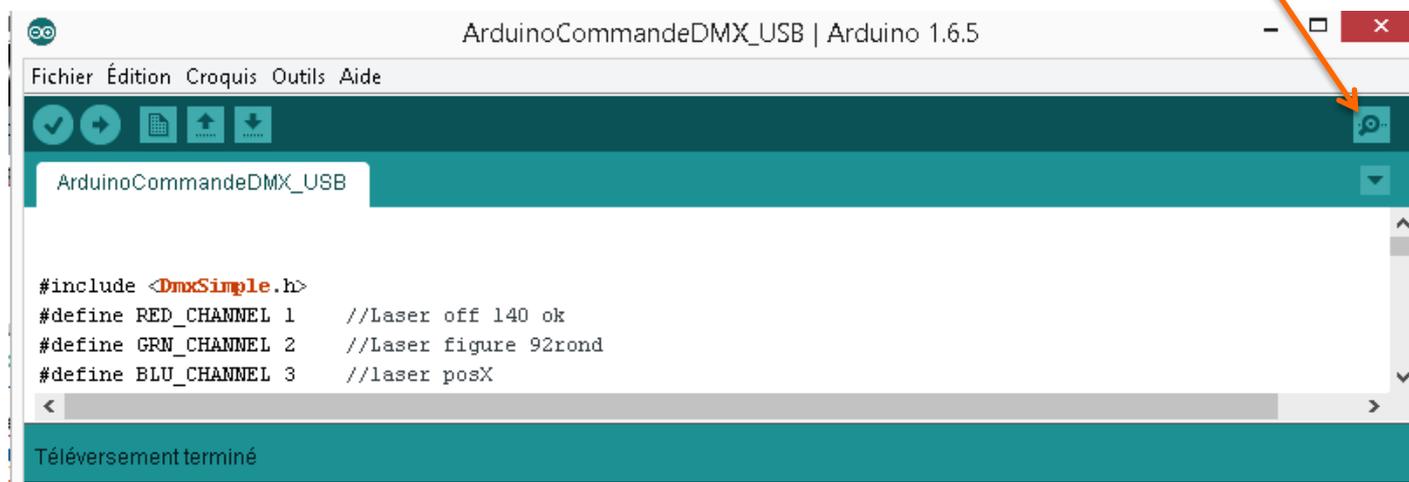
Générer des tâches de couleur en dosant chaque composantes Rouge, Vert, Bleu.

Implanter le programme `ArduinoCommandeDMX_USB.ino` dans la carte Arduino.

Connecter la carte Arduino [nano ou UNO –secours-] au Spot DMX

Commande du spot numérique :

Utiliser l'icône "LOUPE" pour passer en mode "MONITORING"



Fenêtre de monitoring pour envoyer les composantes de couleur :



Bien vérifier que l'option Retour chariot est activée

La fenêtre de monitoring permet de moduler les niveaux de rouge, vert et bleu du spot via une valeur :

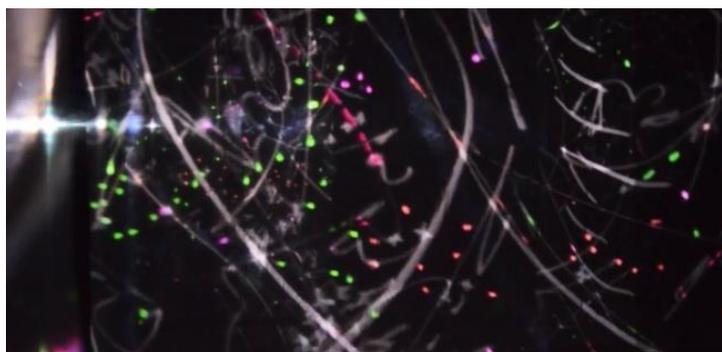
- Moduler le niveau lumineux composante rouge d'un spot [commande R]
- Moduler le niveau lumineux composante vert d'un spot [commande V]
- Moduler le niveau lumineux composante bleu d'un spot [commande B]

Ex :Envoyer avec le Monitoring des messages texte du type :

- R0
- V180
- B50

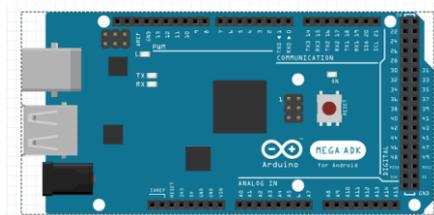


Faire des tests pour régénérer des couleurs composées dans l'esprit du travail de l'équipe de Pascal Dusapin, Thierry Coduys et Guillaume Jacquemin.



Remarque :

Pour implanter le programme dans la carte Arduino il faut via le menu outil sélectionner le type de carte utilisée [nano ou UNO] et sélectionner le port série connecté au PC.



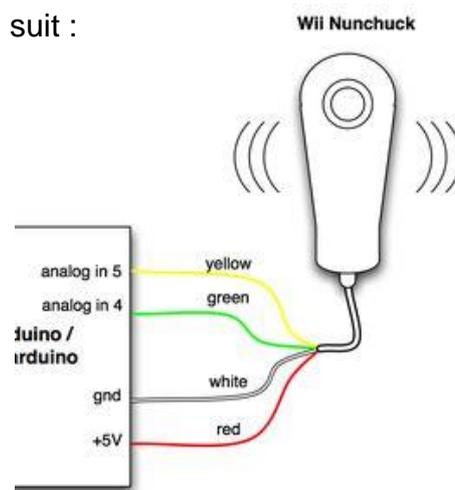
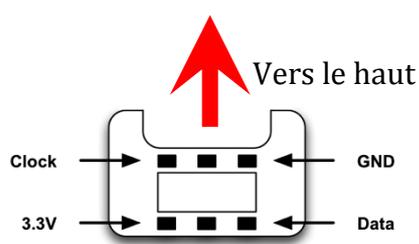
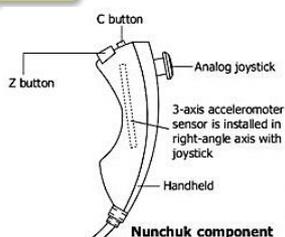
Implanter le programme `ArduinoCommandeSpotViaNunchuck.ino` dans la carte Arduino.

Connecter la carte Arduino à la manette Nunchuck [si partie Shuntée utiliser une carte MEGA ADK]

Manette de jeu Nunchuck

Connecter la manette de jeu nunchuck sur le connecteur comme suit :

+: 3,3V
- : Gnd
d : a4
c : a5



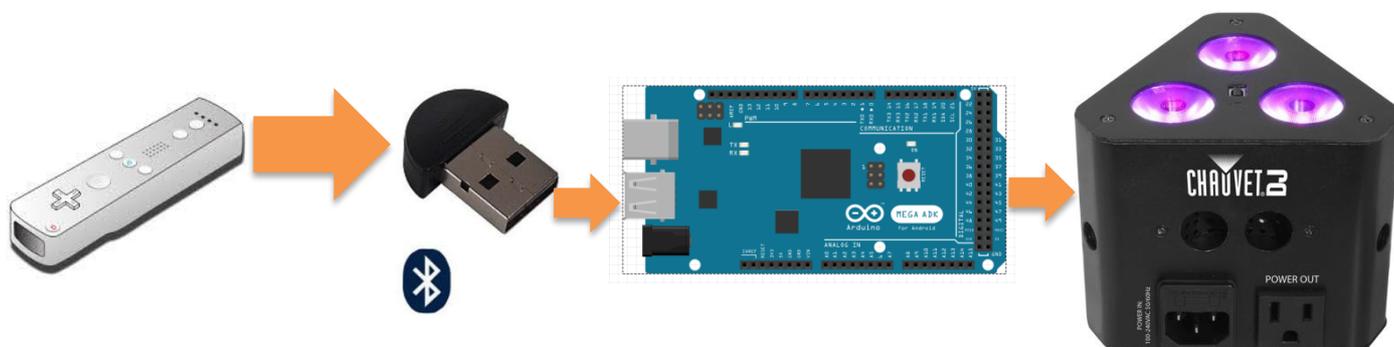
Fonctionnement :

- Si appui bouton Z et pas appui bouton C alors modification du niveau de rouge
- Si pas appui bouton Z et appui bouton C alors modification du niveau de vert (à modifier)
- Si appui bouton Z et appui bouton C alors modification du niveau de bleu (à modifier)

Tester ce dispositif de modulation lumineuse à commande Nunchuck

Commande du SPOT DMX via manette Wiimote

Envoi des commandes numériques du spot scénique par connexion [DMX] pour régler le niveau de rouge, de vert et de bleu présents dans la lumière émise.
L'envoi de commandes numériques sera effectué via une liaison Bluetooth sans fil pour moduler la lumière produite par le spot.



Implanter le programme `ArduinoCommandeSpotViaWiimote.ino` dans la carte Arduino.

Modifier le code de manière à gérer les composantes rouge, verte et bleue par la manette Wiimote.

La manette dispose des boutons LEFT, RIGHT, UP, DOWN, PLUS, MINUS...

```
ArduinoCommandeSpotViaWiimote | Arduino 1.6.5
Fichier Édition Croquis Outils Aide
ArduinoCommandeSpotViaWiimote
}
else {
  if (Wii.getButtonClick(LEFT)) {
    Wii.setLedOff();
    Wii.setLedOn(LED1);
    /*****APPUI BOUTON LEFT*****/
    bleu=bleu-20;
    bleu=constrain(bleu, 0, 255);
    DmxSimple.write(BLU_CHANNEL, bleu);
    Serial.println(bleu);
    Serial.print(F("\r\nLeft"));
    /*****/
  }
  if (Wii.getButtonClick(RIGHT)) {
    Wii.setLedOff();
    Wii.setLedOn(LED3);
    /*****APPUI BOUTON RIGHT*****/
    bleu=bleu+20;
    bleu=constrain(bleu, 0, 255);
    DmxSimple.write(BLU_CHANNEL, bleu);
    Serial.println(bleu);
    Serial.print(F("\r\nRight"));
    /*****/
  }
}
```

Tester votre programme dans l'esprit de l'installation de l'équipe de Pascal Dusapin, Thierry Coduys et Guillaume Jacquemin.

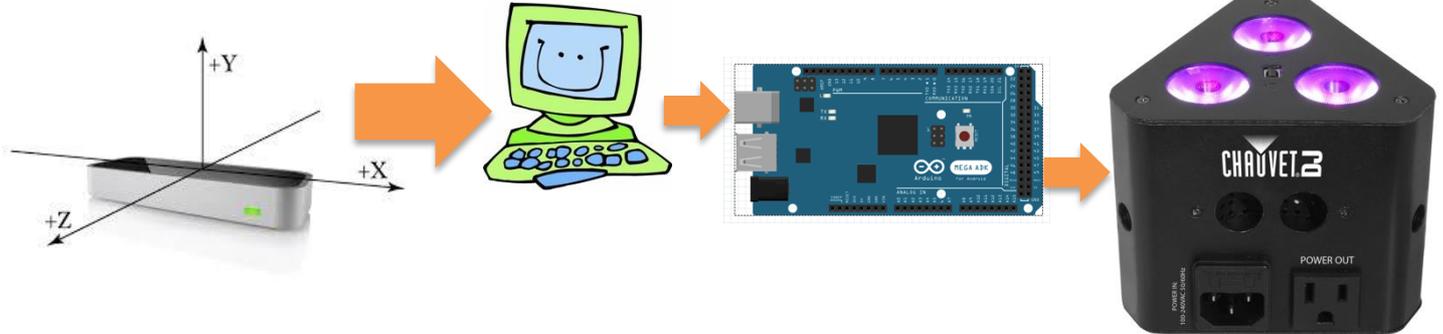


Commande du SPOT DMX via Leapmotion

Envoi des commandes numériques du spot scénique par connexion [DMX] pour régler le niveau de rouge, de vert et de bleu présents dans la lumière émise.

L'envoi de commandes numériques sera effectué via un ordinateur et une carte Arduino pour moduler la lumière produite par le spot.

Le programme ArduinoCommandeDMX_USB.ino est implanté dans l'arduino.



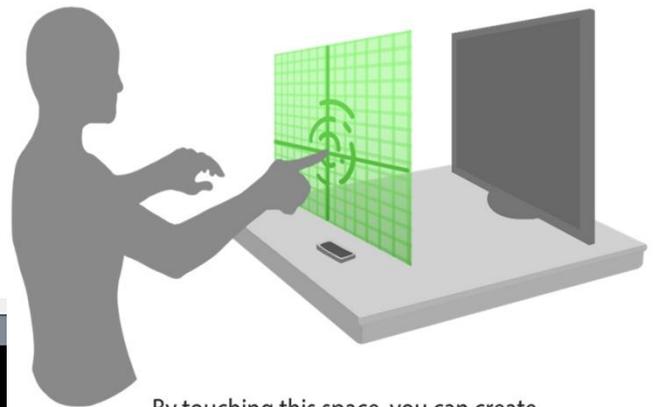
Implanter le programme ProcessingCommandeSpotViaLeapmotion.pde dans l'ordinateur.

```
ProcessingCommandeSpotViaLeapmotion | Processing 2.2.1
File Edit Sketch Tools Help
ProcessingCommandeSpotViaLeapmotion EnvoiCmdDMX Java
int composanteB=0;

/*public*/
void setup() {
  size(1000, 500);
  leap = new LeapMotionP5(this);
  println(Serial.list());
  String nomDesPortSerieDetecte = Serial.list()[0]; // changer 0 par le numéro du port de l'Arduino...
  monPortamoioi = new Serial(this, nomDesPortSerieDetecte, 9600); // ouvre l eport série à 9600bit/s
}

/*public*/
void draw()
{
  background(0);
  fill(0, 255, 0);
  for (Finger finger : leap.getFingerList())
  {
    PVector fingerPos = leap.getTip(finger);
    ellipse(fingerPos.x, fingerPos.y, 10, 10);
    println("posX="+fingerPos.x+" posY="+fingerPos.y+" posZ="+fingerPos.z );
    composanteR=int (map (fingerPos.x,200,800,0,255));
    composanteR=constrain (composanteR,0,255);
    composanteV=int (map (fingerPos.y,400,150,0,255));
    composanteV=constrain (composanteV,0,255);
    composanteB=int (map (fingerPos.z,200,-200,0,255));
    composanteB=constrain (composanteB,0,255);
    println("R="+composanteR+" V="+composanteV+" B="+composanteB );
    Emission();
  }
}
<
Done Saving.
R=186 V=0 B=229
posX=450.89124 posY=459.829 posZ=-163.85277
R=186 V=0 B=231
posX=451.17398 posY=461.29938 posZ=-163.69505
R=186 V=0 B=231
posX=450.68257 posY=461.88347 posZ=-159.84581
R=186 V=0 B=229
posX=450.54117 posY=462.637 posZ=-154.98792
R=186 V=0 B=226
1
```

The Leap Motion Controller can be used to create a virtual 'touch' surface in the air.



By touching this space, you can create virtual touch events on your screen

SLASH GEAR

Next ▶

Modifier le code pour moduler la couleur de la tâche de lumière en fonction de la gestuelle devant le capteur. Par exemple choix de l'axe qui module chacune des couleurs RVB..

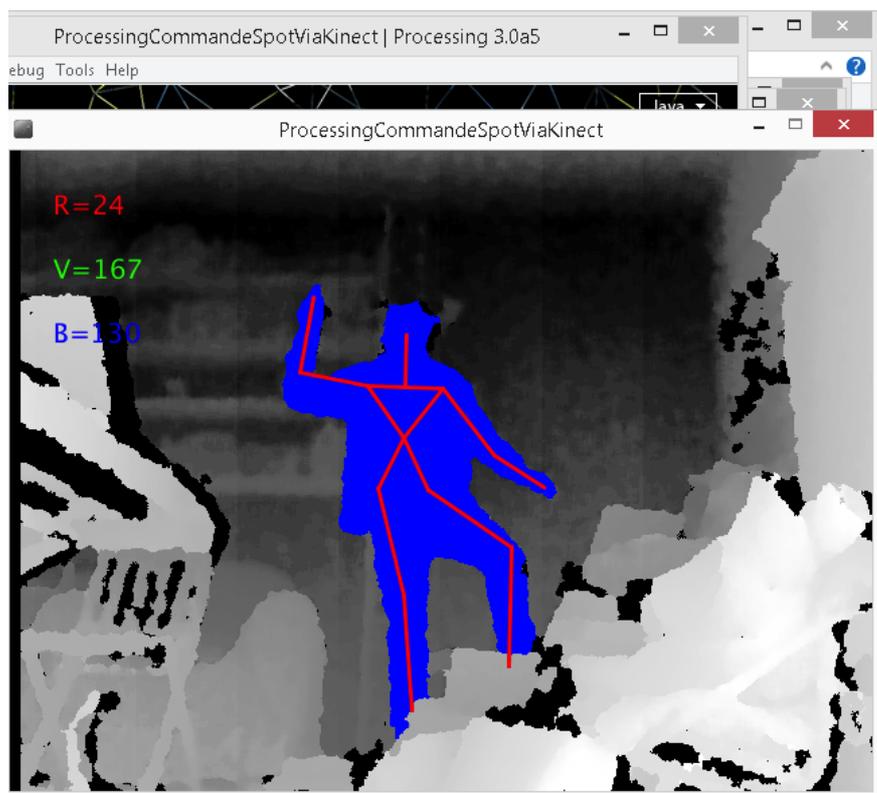
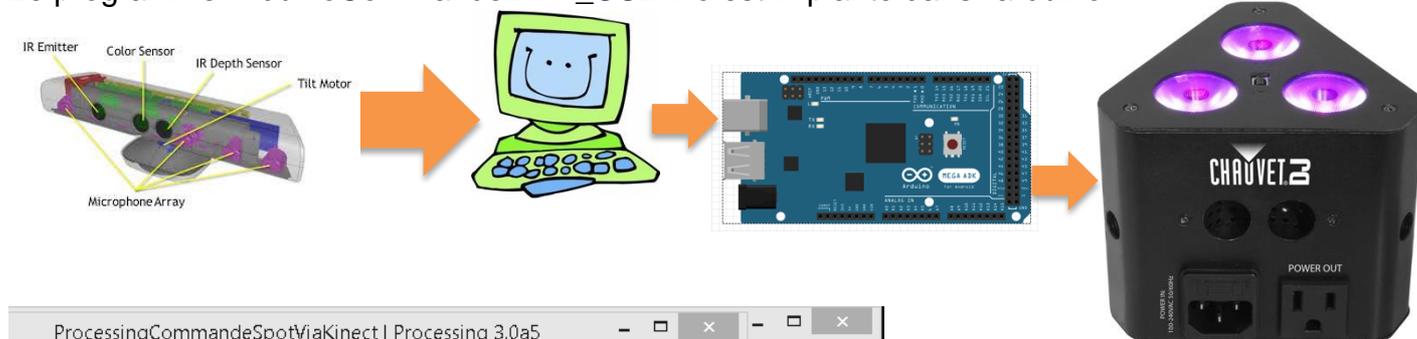
Tester votre programme dans l'esprit de l'installation de l'équipe de Pascal Dusapin, Thierry Coduys et Guillaume Jacquemin.



Commande du SPOT DMX via Kinect

Envoi des commandes numériques du spot scénique par connexion [DMX] pour régler le niveau de rouge, de vert et de bleu présents dans la lumière émise.
L'envoi de commandes numériques sera effectué via un ordinateur et une carte Arduino pour moduler la lumière produite par le spot.

Le programme `ArduinoCommandeDMX_USB.ino` est implanté dans l'arduino



Implanter le programme `ProcessingCommandeSpotViaKinect.pde` dans l'ordinateur.

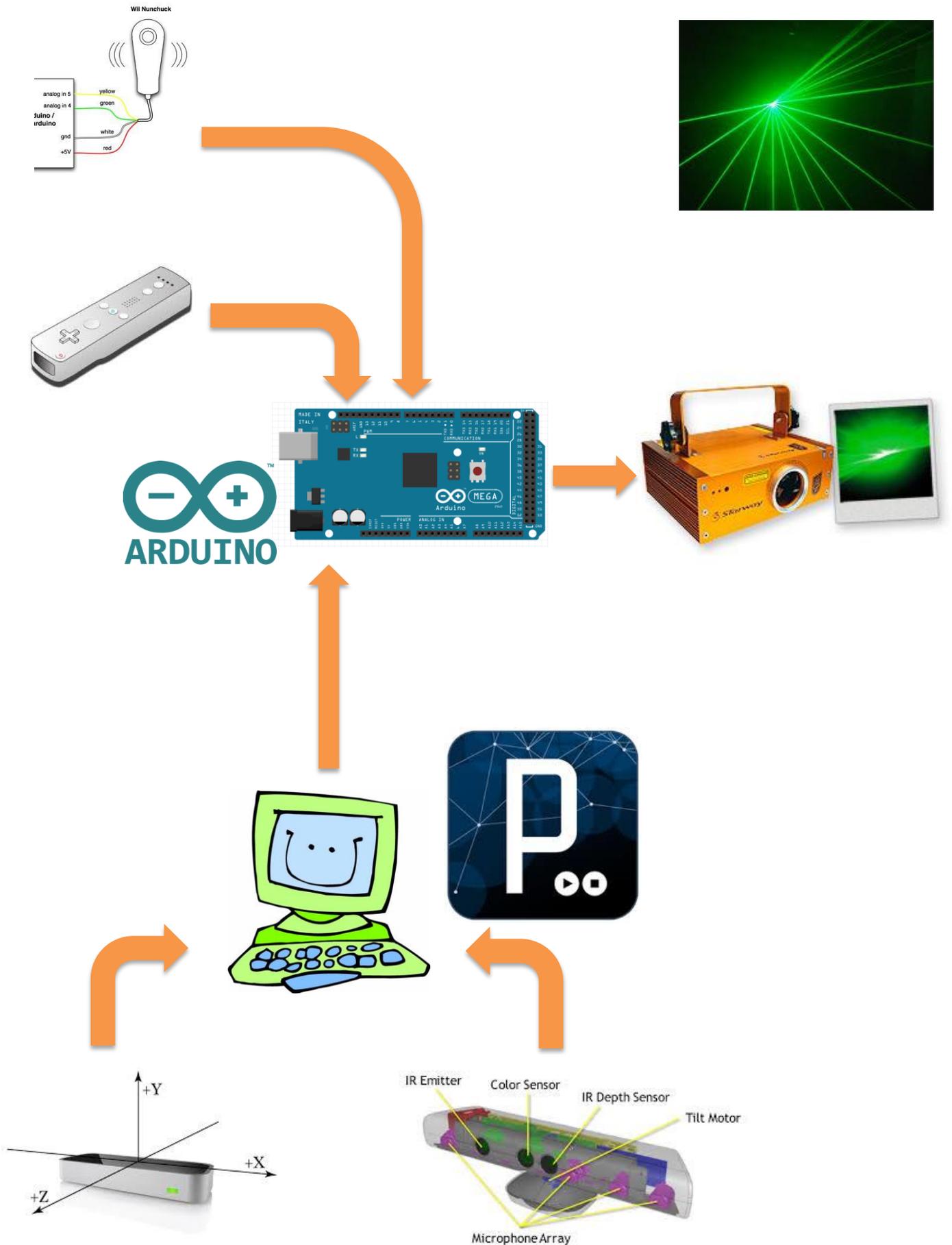
Modifier le code pour moduler la couleur de la tâche de lumière en fonction de la gestuelle devant le capteur. S'inspirer des lignes de code générant la composant de couleur rouge.

- Modulation du niveau de rouge avec la main droite

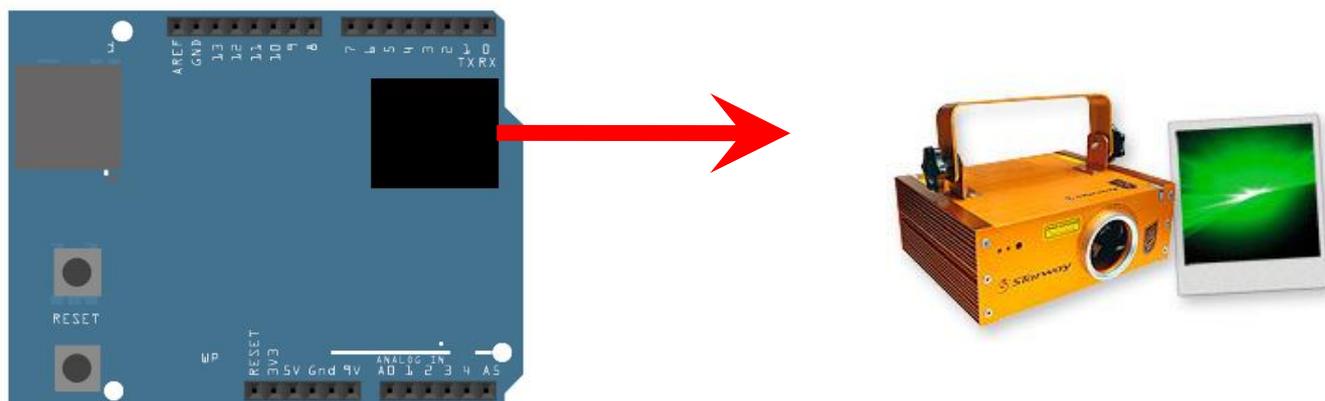
Tester votre programme dans l'esprit de l'installation de l'équipe de Pascal Dusapin, Thierry Coduys et Guillaume Jacquemin.



Activité Lumière 2 : Projeter des traits de lumière



Commande du LASER



Le laser dispose d'une connexion numérique [DMX] qui permet de régler la position du faisceau de lumière.

L'envoi de commandes numériques permet de déplacer le faisceau.

Besoin technique :

Un PC

Le logiciel de développement Arduino

Une carte électronique programmable Arduino

Une manette de jeu Nunchuck.

Un laser numérique [à commande DMX]

Fiche activité :

Le déplacement d'un faisceau laser peut reproduire les constructions graphiques de Pascal Dusapin.

Les élèves réalisent le codage de la position en paramétrant les coordonnées X, Y du faisceau émis.

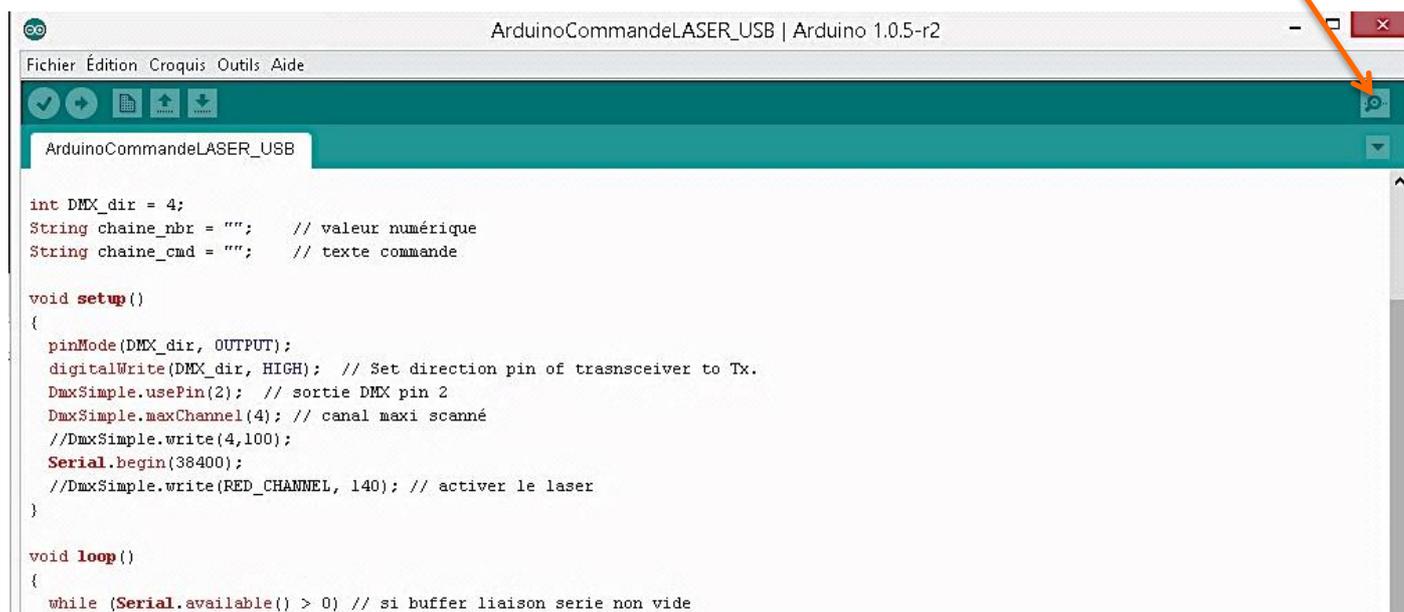


Activité :

Ouvrir le logiciel Arduino et lancer le programme " ArduinoCommandeLASER_USB.ino "
Déplacer le faisceau laser en changeant les coordonnées X, Y.

Commande du Laser numérique :

Utiliser l'icône "LOUPE" pour passer en mode "MONITORING"



```
ArduinoCommandeLASER_USB | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide
ArduinoCommandeLASER_USB
int DMX_dir = 4;
String chaine_nbr = ""; // valeur numérique
String chaine_cmd = ""; // texte commande

void setup()
{
  pinMode(DMX_dir, OUTPUT);
  digitalWrite(DMX_dir, HIGH); // Set direction pin of transceiver to Tx.
  DmxSimple.usePin(2); // sortie DMX pin 2
  DmxSimple.maxChannel(4); // canal maxi scanné
  //DmxSimple.write(4,100);
  Serial.begin(38400);
  //DmxSimple.write(RED_CHANNEL, 140); // activer le laser
}

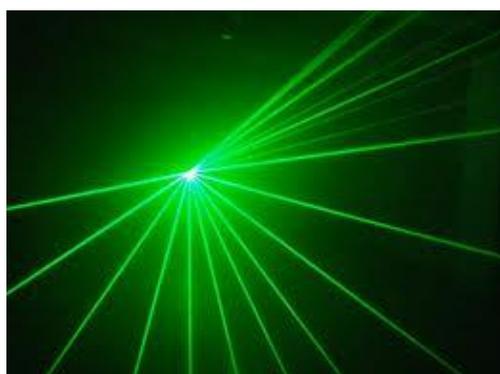
void loop()
{
  while (Serial.available() > 0) // si buffer liaison serie non vide
```

Fenêtre de monitoring pour envoyer les commandes au LASER :



La fenêtre de monitoring permet de moduler les coordonnées du laser via les valeurs X et Y :

- Activer le laser [commande M140 pour ON], [commande M5 pour OFF] ou [M12 animation].
- Sélectionner une figure [commande F] (voir tableau des figures page suivante).
- Déplacer le faisceau laser sur l'axe X [commande X]
- Déplacer le faisceau laser sur l'axe Y [commande Y]
- Sélectionner la taille des figures [commande T]

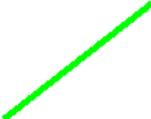
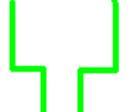


Ex : Envoyer avec le Monitoring des messages texte du type :

- M140
- X100
- F92
- T5
- Y128

Faire des tests pour déplacer le laser.

Liste des figures disponibles :

0-5	6-11	12-17	18-23	24-29
				
30-35	36-41	42-47	48-53	54-59
				
60-65	66-71	72-77	78-83	84-89
				
90-95	96-101	102-107	108-113	114-119
				
120-125	126-131	132-137	138-143	144-149
				
150-155	156-161	162-167	168-173	174-179
				
180-185	186-191	192-197	198-203	204-209
				
210-215	216-221	222-227	228-233	234-239
				
240-245	246-255			
				

Commande du faisceau laser par une manette de jeux Nunchuck

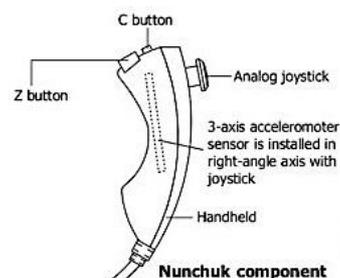
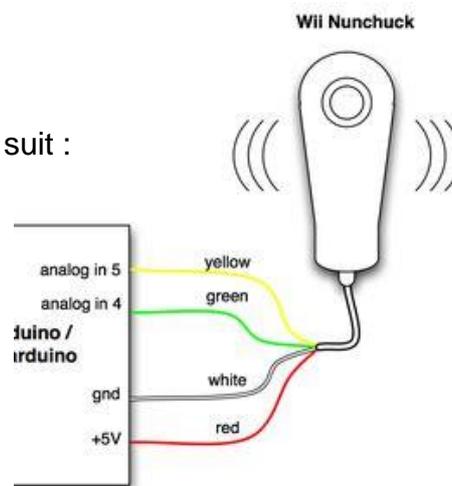
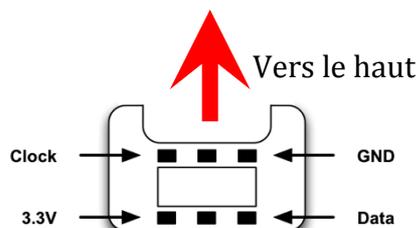
Connecter la manette de jeu à la carte Arduino.

Implanter le programme `ArduinoCommandeLASER_Nunchuck.ino` dans la carte Arduino.
Connecter la carte Arduino à la manette Nunchuck

Manette de jeu Nunchuck

Connecter la manette de jeu Nunchuck sur le connecteur comme suit :

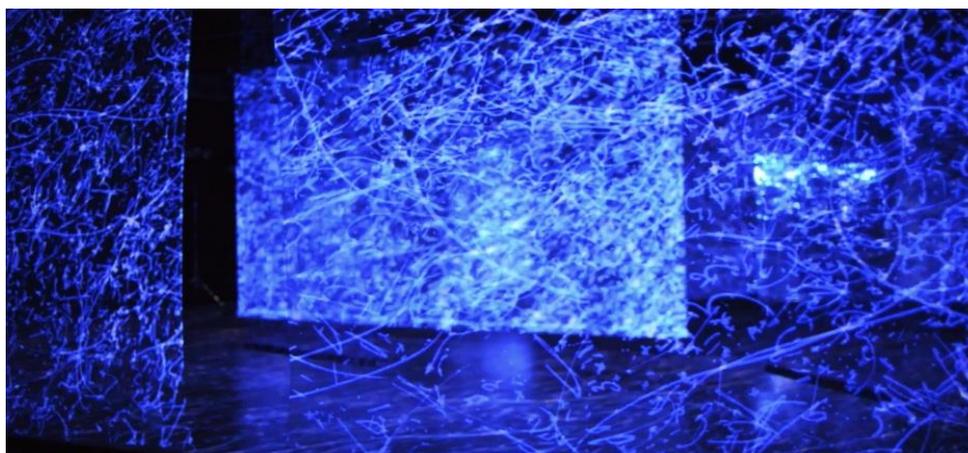
+ : 3,3V
- : Gnd
d : a4
c : a5



Fonctionnement :

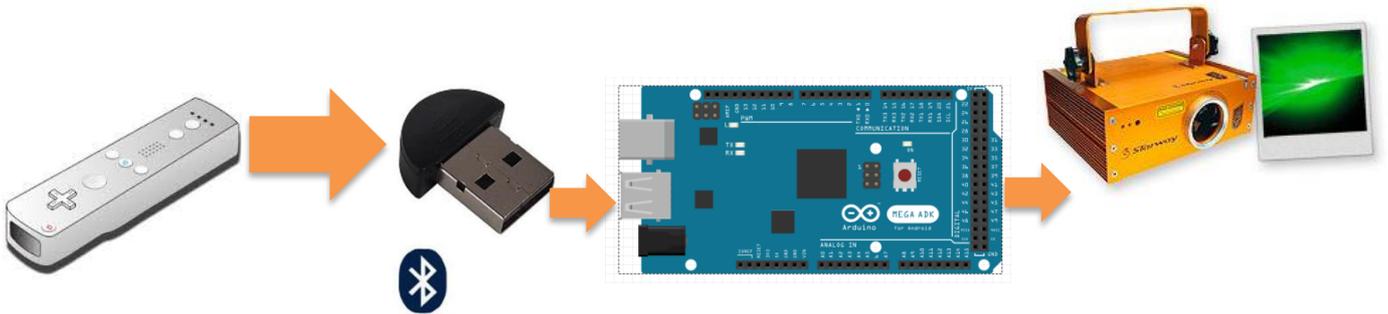
- La position horizontale du joystick modifie la position du faisceau sur l'axe X
-
- La position verticale du joystick modifie la position du faisceau sur l'axe Y (à modifier)

Tester ce dispositif de commande du laser à la manière de l'équipe de Pascal Dusapin, Thierry Coduys et Guillaume Jacquemin



Commande du faisceau LASER via manette Wiimote [+ Nunchuck...]

Envoi des commandes numériques au laser scénique par connexion [DMX] pour régler la position du faisceau de lumière.
L'envoi de commandes numériques sera effectué via une liaison Bluetooth sans fil pour déplacer le faisceau.



Implanter le programme `ArduinoCommandeLASERViaWiimote.ino` dans la carte Arduino.

Modifier le code de manière à gérer les déplacements, figure et l'activité du laser par la manette Wiimote.

La manette dispose des boutons LEFT, RIGHT, UP, DOWN, PLUS, MINUS, A, B...

Ex : code pour activer – désactiver le laser.

```
ArduinoCommandeLASERViaWiimote | Arduino 1.6.5
Fichier Édition Croquis Outils Aide
ArduinoCommandeLASERViaWiimote $
  Serial.print(F("\r\nTwo"));

  if (Wii.getButtonClick(A))
  {
    printAngle = !printAngle;
    /*****APPUI BOUTON A*****/
    DmxSimple.write(MOD_CHANNEL, 140); // activer le laser
    Serial.println("Activation du laser");
    Serial.print(F("\r\nA"));
    /*****/
  }
  if (Wii.getButtonClick(B))
  {
    /*****APPUI BOUTON B*****/
    DmxSimple.write(MOD_CHANNEL, 5); // désactiver le laser
    Serial.println("Désactivation du laser");
    //Wii.setRumbleToggle();
    Serial.print(F("\r\nB"));
    /*****/
  }
}

Compilation terminée.

ArduinoCommandeLASERViaWiimoteRef
  /*****APPUI BOUTON LEFT*****/
  posX=posX-25;
  posX=constrain(posX, 0, 255);
  DmxSimple.write(POX_CHANNEL, posX);
  Serial.println(posX);
  Serial.print(F("\r\nLeft"));
  /*****/
}
if (Wii.getButtonClick(RIGHT)) {
  Wii.setLedOff();
  Wii.setLedOn(LED3);
  /*****APPUI BOUTON RIGHT*****/
  posX=posX+25;
  posX=constrain(posX, 0, 255);
  DmxSimple.write(POX_CHANNEL, posX);
  Serial.println(posX);
  Serial.print(F("\r\nRight"));
  /*****/
}
}
```

Positionnement axes X et Y :

Boutons Left et Right de la Wiimote

ANNEXE :

Atelier de création du 26, 27 et 28 avril 2016

FICHE n°1 : FORME (=STRUCTURE) POSSIBLE POUR LA CREATION

* une forme ternaire (en 3 parties)



> la partie A revient en fin de création de manière identique

> la partie A peut être différente de la partie B par sa vitesse, son volume sonore, son atmosphère, ses lumières...



> la dernière partie A' est très ressemblante à la première partie MAIS peut se différencier par l'utilisation d'un sample différent, par une vitesse plus réduite, par l'utilisation moins nombreuses de notes, par un volume sonore plus important ...

* une forme couplet / refrain (à la manière d'une chanson)



> les parties A1 et A2 sont des couplets, c'est à dire qu'elles se ressemblent mais ne sont pas pour autant identique (par exemple : la partie musicale est identique entre la partie A1 et la partie A2 mais la mise en lumière est différente)

> les parties refrain sont identiques entre elles

FICHE n°2 : CONSIGNES > autour de la GAMME PENTATONIQUE Do min

Avant de pouvoir coder les notes voulues, comme cela avait été réalisé au mois de JANVIER au Lieu Unique

Autour du piano :

- * repérer les gommettes de couleurs
- * jouer toutes les notes de la gamme avec les gommettes de couleurs
- * essayer d'associer des notes ensemble, de créer des petites mélodies.

Autour du tableau des fréquences :

- * repérer dans le tableau des fréquences, (ci-dessous) les notes correspondantes à la gamme de Do min
- * faites un choix sur les hauteurs voulues pour ces différentes notes : GRAVE /MEDIUM / AIGU
- * une fois votre choix effectué, reporter les fréquences choisies sous les notes au niveau de

CHOIX Correspondance Hertz

Tableau de correspondance NOTES > FREQUENCE

Hauteur	GRAVE			MEDIUM		AIGU		
	0	1	2	3	4	5	6	7
DO	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
Mi b	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03
FA	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
FA#	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
SOL	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
SI b	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62

Recherche de mélodies autour de la GAMME PENTATONIQUE

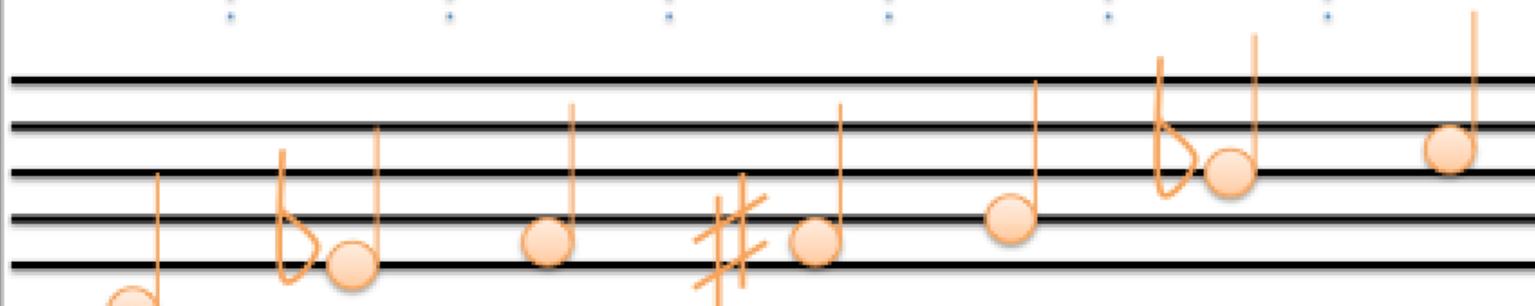
Do min

Notes



DO Mib FA (FA#) SOL Sib DO

Représentation
Graphique



Couleurs



CHOIX

Correspondance
Hertz



Do Mib Fa Fa# Sol Sib Do

FICHE n°3 : CONSIGNES > autour de la GAMME PENTATONIQUE Ré min

Avant de pouvoir coder les notes voulues, comme cela avait été réalisé au mois de JANVIER au Lieu Unique

Autour du piano :

- * repérer les gommettes de couleurs
- * jouer toutes les notes de la gamme avec les gommettes de couleurs
- * essayer d'associer des notes ensemble, de créer des petites mélodies.

Autour du tableau des fréquences :

- * repérer dans le tableau des fréquences, (ci-dessous) les notes correspondantes à la gamme de Ré min
- * faites un choix sur les hauteurs voulues pour ces différentes notes : GRAVE /MEDIUM / AIGU
- * une fois votre choix effectué, reporter les fréquences choisies sous les notes au niveau de



CHOIX Correspondance Hertz

Tableau de correspondance NOTES > FREQUENCE

Hauteur	GRAVE			MEDIUM		AIGU		
	0	1	2	3	4	5	6	7
RE	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64
FA	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
SOL	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
SOL #	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
LA	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00
DO	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01

Recherche de mélodies autour de la GAMME PENTATONIQUE

Ré min

Notes



RE

FA

SOL

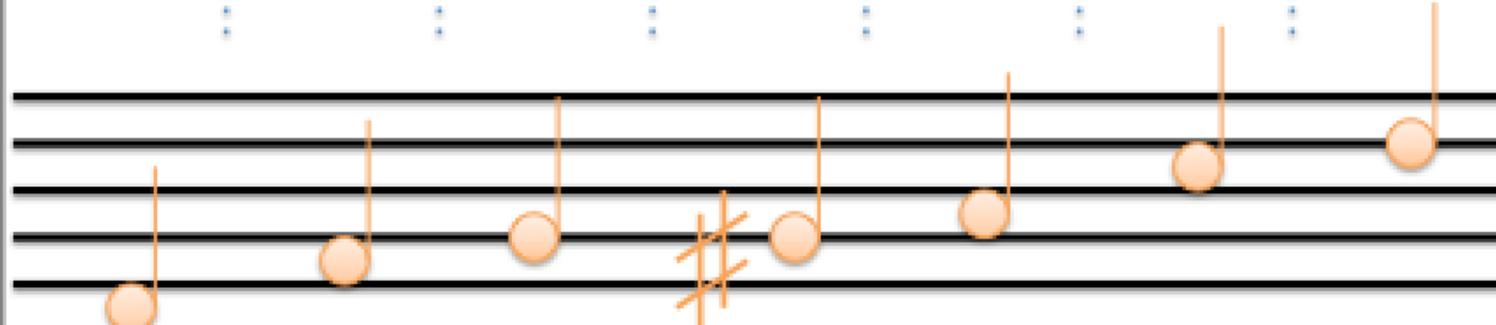
(SOL#)

LA

DO

RE

Représentation
Graphique



Couleurs



CHOIX

Correspondance
Hertz



Ré

Fa

Sol

Sol#

La

Do

Ré

FICHE n°4 : CONSIGNES > autour de la GAMME PENTATONIQUE MI Maj

Avant de pouvoir coder les notes voulues, comme cela avait été réalisé au mois de JANVIER au Lieu Unique

Autour du piano :

- * repérer les gommettes de couleurs
- * jouer toutes les notes de la gamme avec les gommettes de couleurs
- * essayer d'associer des notes ensemble, de créer des petites mélodies.

Autour du tableau des fréquences :

- * repérer dans le tableau des fréquences, (ci-dessous) les notes correspondantes à la gamme de Mi Maj
- * faites un choix sur les hauteurs voulues pour ces différentes notes : GRAVE /MEDIUM / AIGU
- * une fois votre choix effectué, reporter les fréquences choisies sous les notes au niveau de

 **CHOIX Correspondance Hertz**

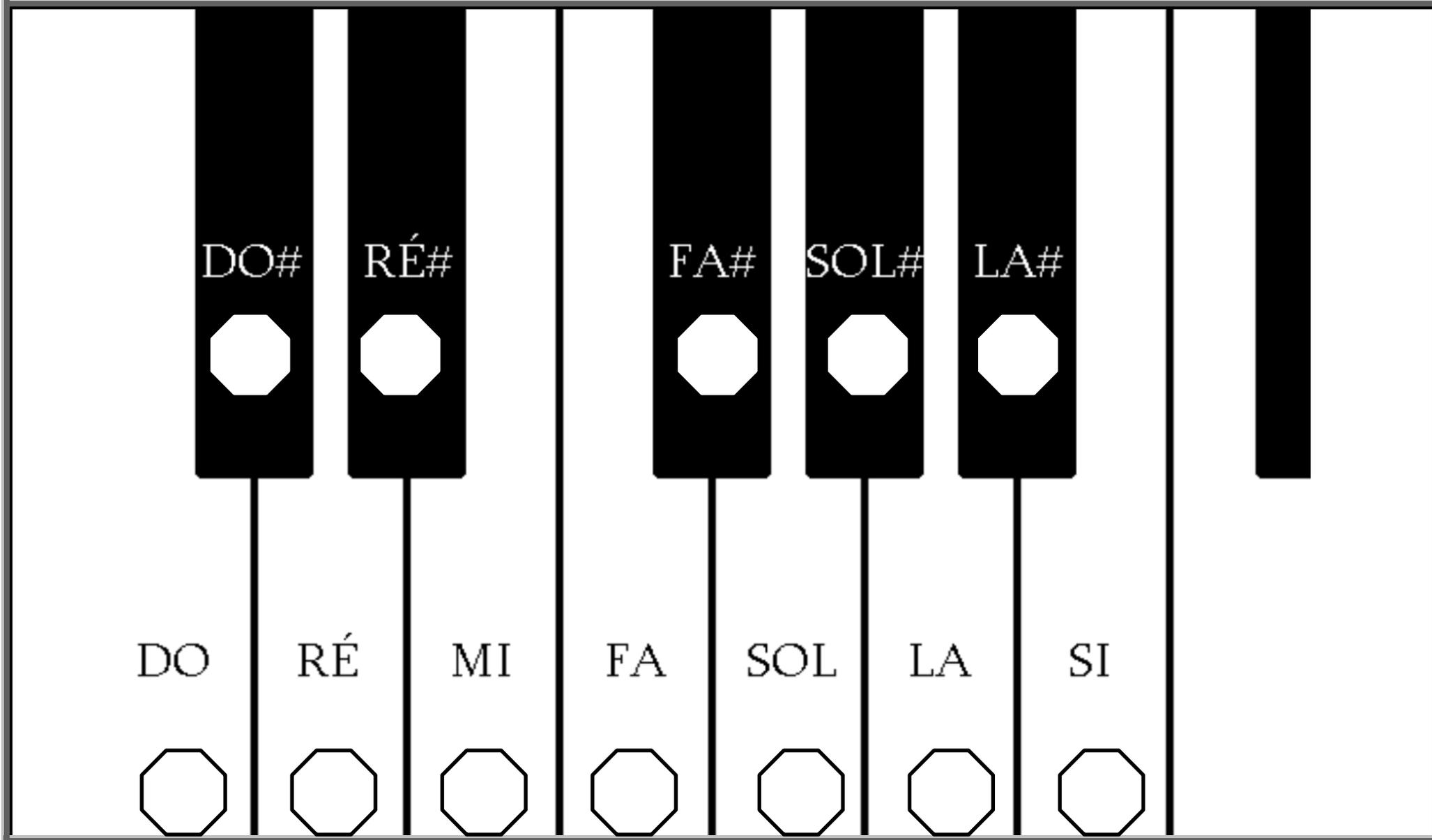
Tableau de correspondance NOTES > FREQUENCE

Hauteur	GRAVE			MEDIUM		AIGU		
Note	0	1	2	3	4	5	6	7
MI	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04
FA#	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
SOL #	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
SI	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13
DO#	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92

Recherche de mélodies autour de la GAMME PENTATONIQUE Mi Maj.

Notes		MI	FA#	SOL#	SI	DO#	MI	aigu (SOL#)
Représentation Graphique								
Couleurs								
CHOIX		Mi	Fa#	Sol#	Si	Do#	Mi	Sol#
Correspondance Hertz								

FICHE n°5 : Recherche de mélodies LIBRES
(sans contrainte de gammes)



CORRESPONDANCE NOTES CHOISIES > FREQUENCE **(Recherche de mélodies LIBRES)**

<u>Note 1 :</u>							
<u>Fréquence :</u>							

Hauteur	GRAVE			MEDIUM		AIGU		
Note	0	1	2	3	4	5	6	7
SI# = DO	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
DO# = <u>REb</u>	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92
RE	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64
RE# = MI b	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03
MI = FA b	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04
FA = MI #	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
FA# = SOL b	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
SOL	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
SOL # = LA b	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
LA	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00
LA# = SI b	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62
SI = DO b	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13

FICHE n°6 : CONSIGNES > CREATION MELODIES

Chaque rond correspond à une note :

- * colorier selon son souhait cette grille avec les couleurs des notes choisies
- * possibilité de jouer 2 notes superposées c'est à dire au même moment (> 2 ronds coloriés superposés), 3 notes superposées (3 ronds coloriés superposés)

Exemple possible :

Do	Do/Mib	Fa	Do	Do/Fa	Mib/Sib/Do	Do	Do/Mib	Fa/Sib	Sib/Mib/Fa
									
									
									

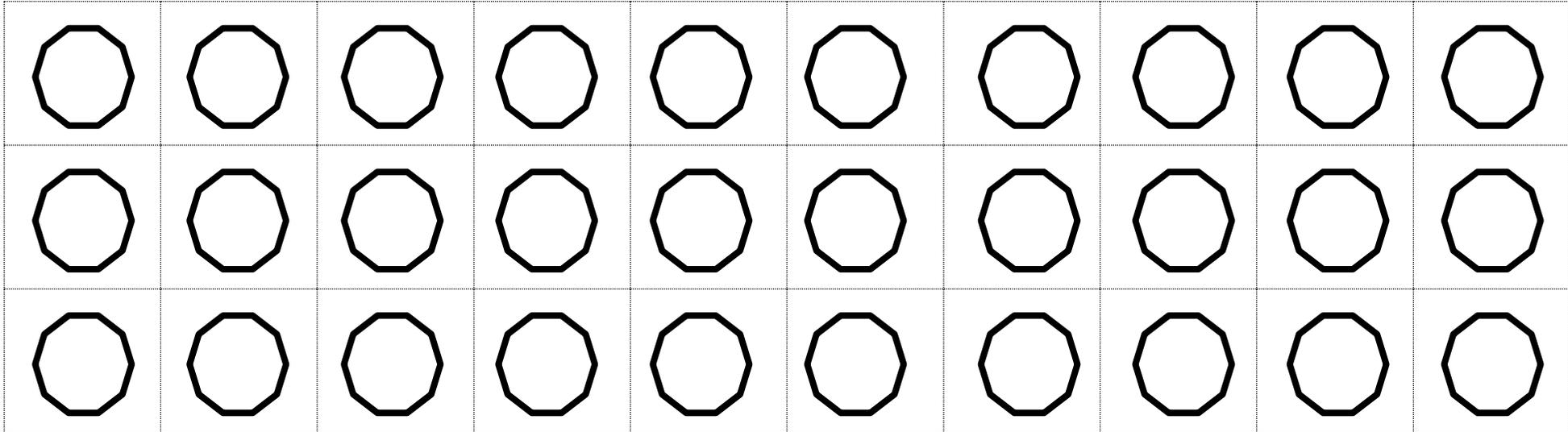
Essayer de créer 3 ou 4 mélodies :

- * en variant les associations de notes
- * en essayant d'ajouter aux notes une rythmique

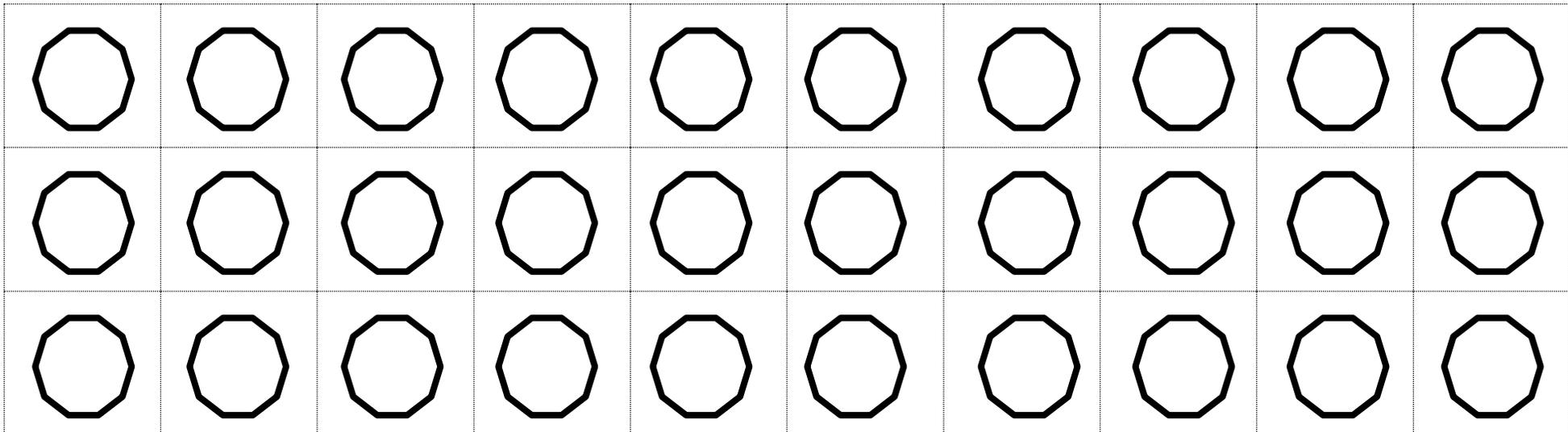
Une fois le choix des mélodies réalisé :

- * coder avec les couleurs des notes correspondantes 3 ou 4 partitions graphiques
- * n'hésiter pas, à l'aide d'un « enregistreur » (type téléphone) à enregistrer vos 3 ou 4 mélodies, afin de les mémoriser plus facilement.

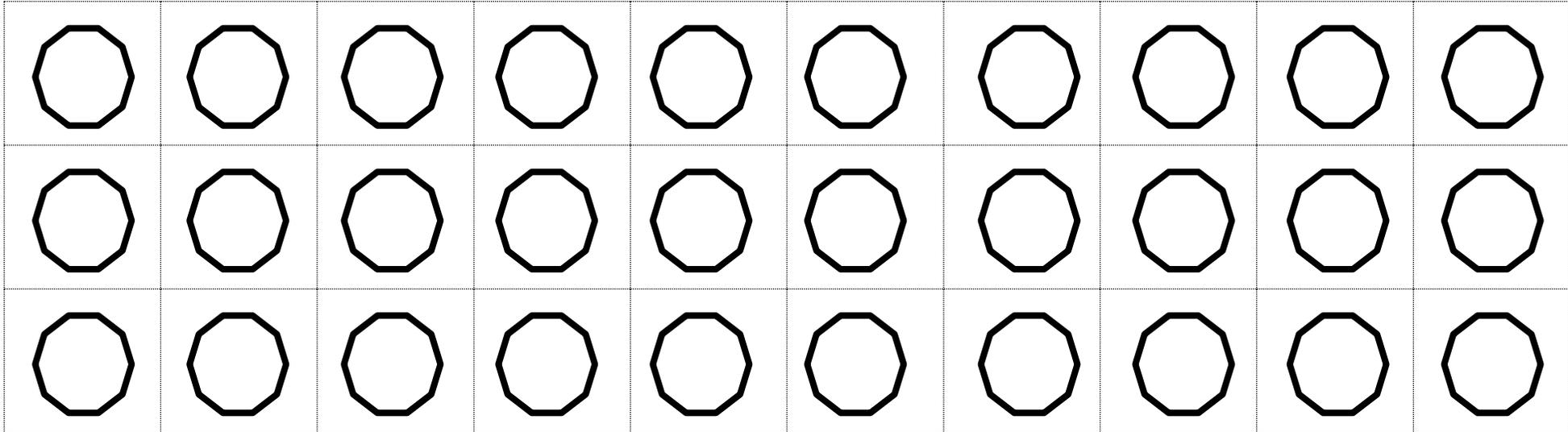
CREATION MELODIE 1



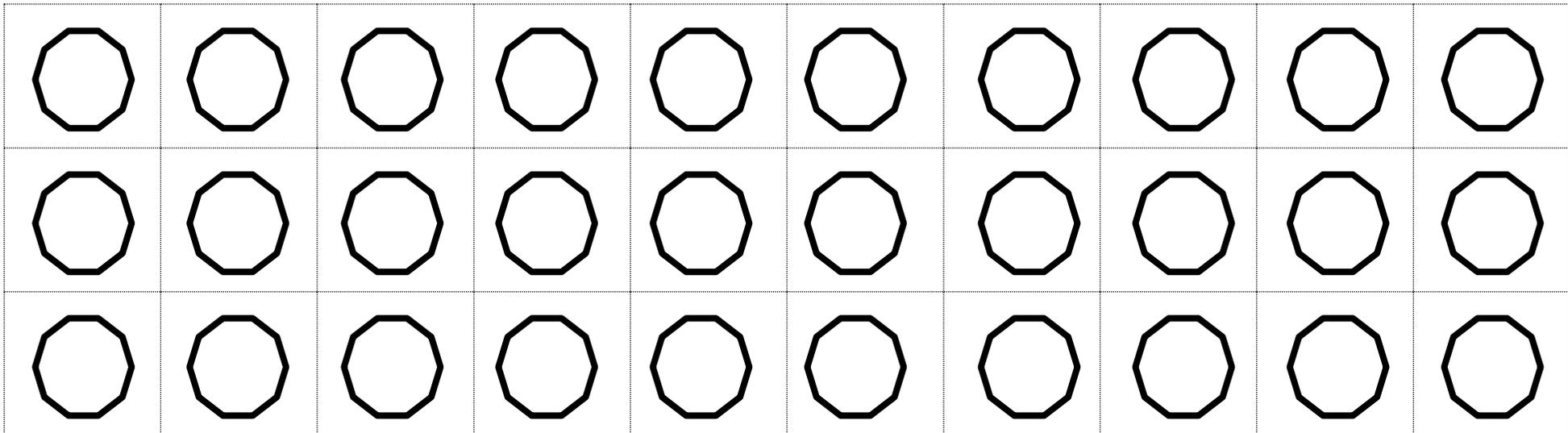
CREATION MELODIE 2



CREATION MELODIE 3



CREATION MELODIE 4



FICHE n°7 : COMMENT TRANSCRIRE CE QUE L'ON ENTEND ou PRODUIT PAR DES CODES GRAPHIQUES ?

Quelques exemples de partitions graphiques

OOO.MOO

POOST YA
THE BURN-
ING BUSH

ODYODZKE
GING

NOT AS A BIRTH
BUT AS LOVE

THAT SHALL

CONTINUE

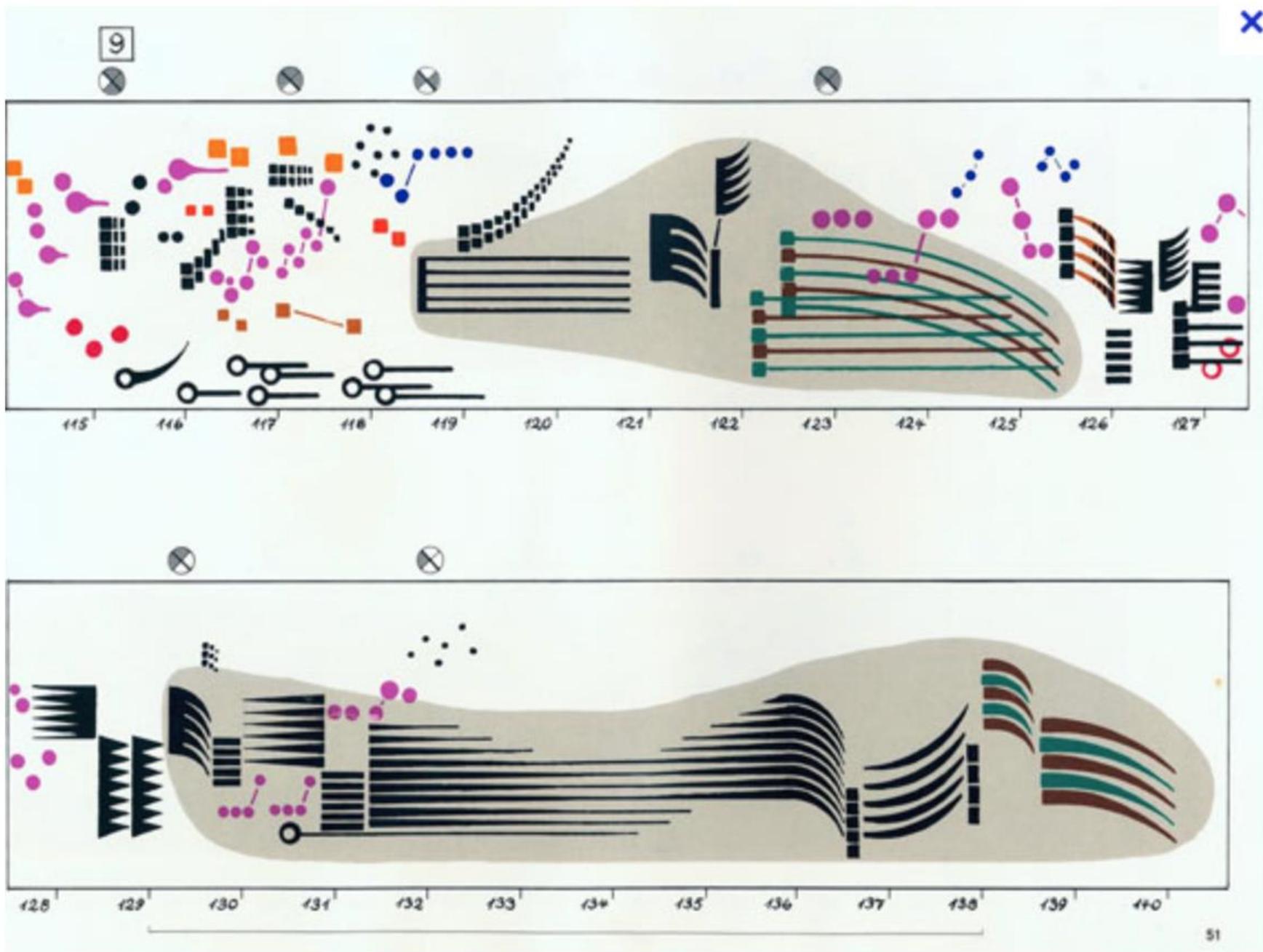
FAR-REMOVED

WILL YOU
GIVE ME TO
TELL YOU?

BREAKING
THE SHELL

AT REST

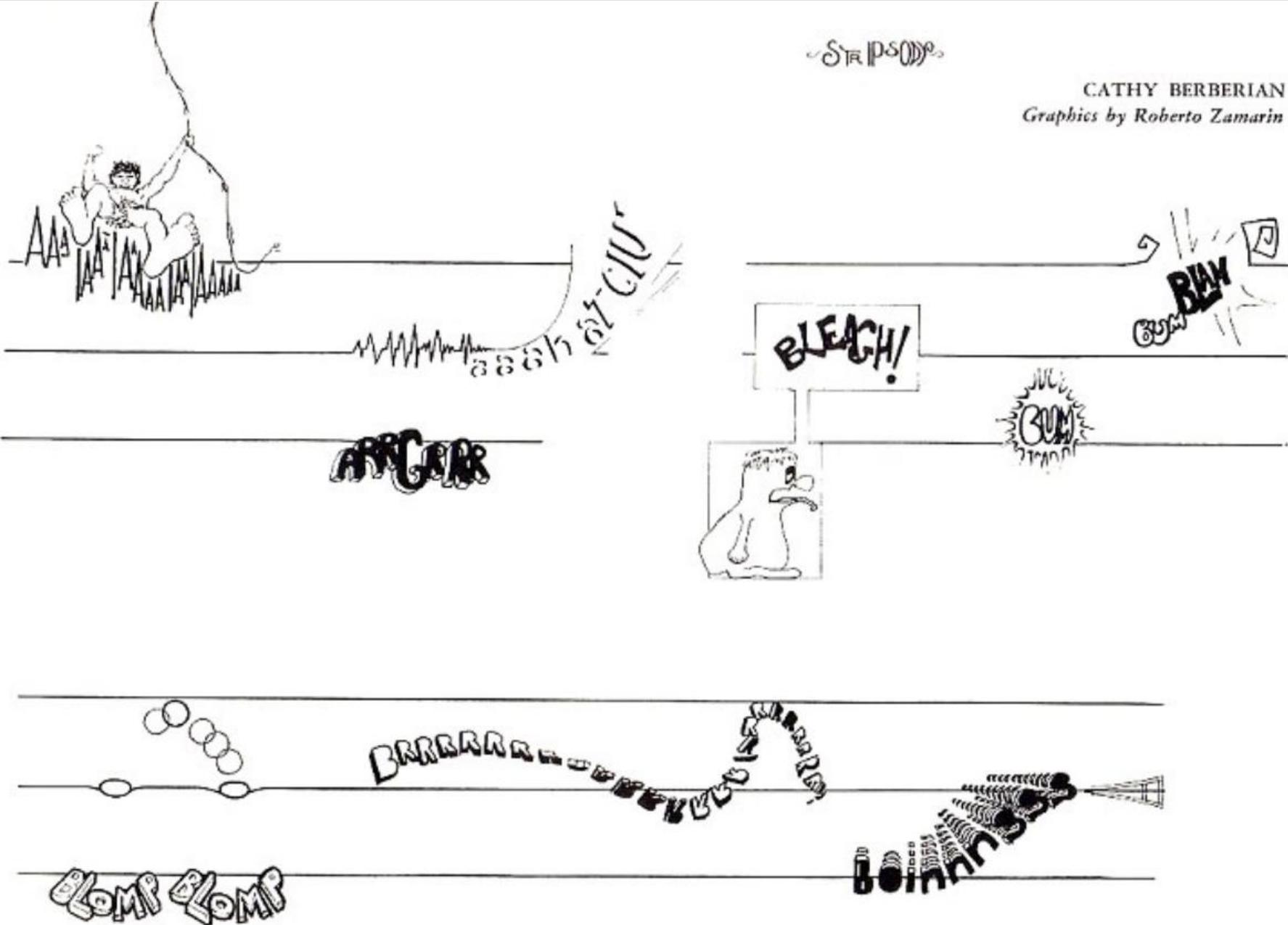
Partition "Aria de John CAGE (1958)



Partition "Artikulation" de György LIGETI (1958) > VIDEO

STRIPSODY

CATHY BERBERIAN
Graphics by Roberto Zamarin

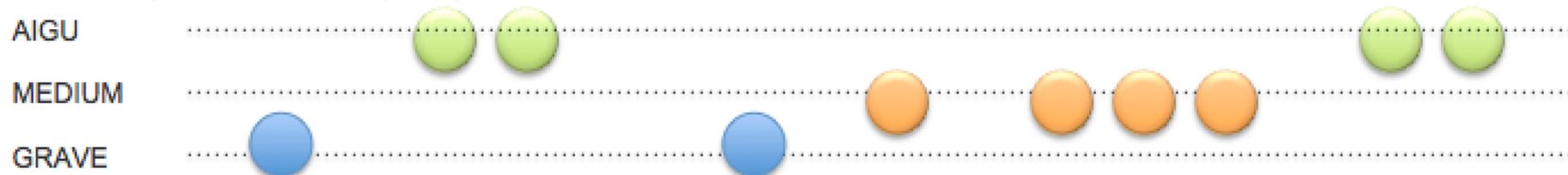


Partition "Stripsody" de et par Cathy BERBERIAN (1966) > VIDEO

COMMENT TRANSCRIRE CE QUE L'ON ENTEND ou PRODUIT PAR DES CODES GRAPHIQUES ?

* avec des couleurs pour différencier les notes codées et donc les sons à jouer

* avec des jeux de hauteurs pour préciser un son GRAVE /MEDIUM/AIGU



* avec des chiffres pour préciser le sample à lancer (1, 2, 3 ...)

* avec des lettres pour différencier des parties avec des atmosphères différentes (A, B, C ... a, b, c ... Couplet 1 / refrain... A1, A2, B1, B2 ...)

* avec des symboles pour par exemple préciser :

- des répétitions X 2, X 3 ...

- des ajouts (= des superpositions) +

- le volume sonore qui est augmenté (=crescendo)

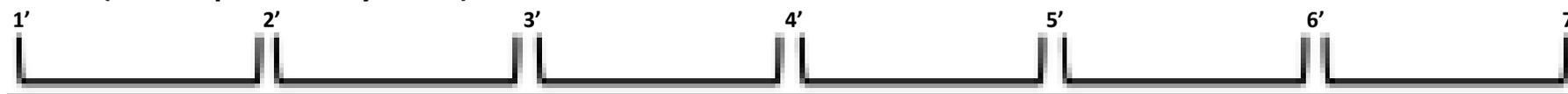


ou bien qui est diminué (=decrescendo)



* avec des dessins pour symboliser un son particulier (la pluie, le vent, l'orage ...)

* avec un échelonnement temporel, permettant ainsi de positionner des sons plus ou moins rapprochés les uns par rapport aux autres (transcription des rythmes).



* ... ou bien avec tout autre idée pouvant au mieux transcrire ce que vous avez créés.

FICHE n°8 : Partition vierge > CODAGE CREATION

TITRE : _____

Page n°

**F
U
M
-
E
R
E**

**S
O
N**

