

LIVRABLE



PROJET IÉTI PIANOCKTAIL

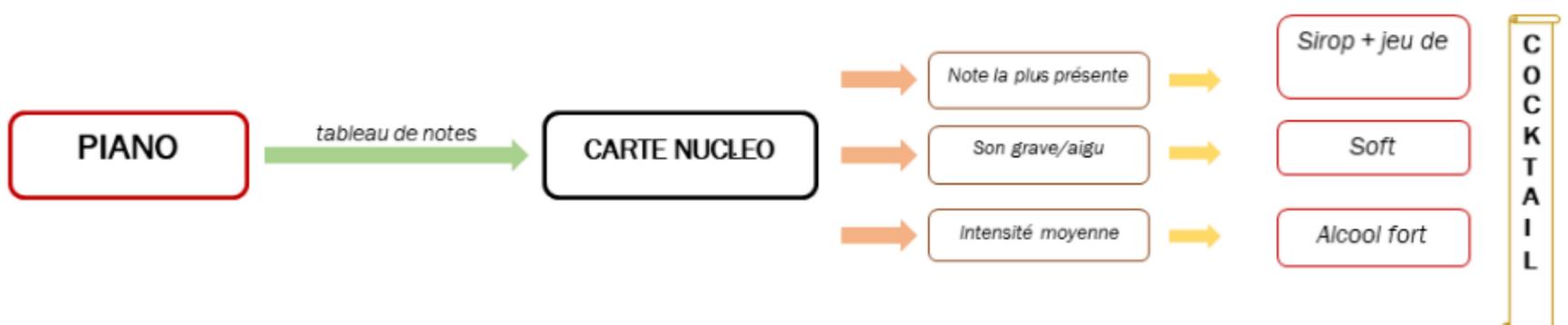
QUI A DIT QUE LA TECHNOLOGIE NE POUVAIT PAS ÊTRE POÉTIQUE ?

Nom du projet	Pianocktail	Période du projet
Personnes en charge du projet	Emma Fevret Françoise Pham Djinar Sadouni Salomé Perrin Rose-Nimol Zacharewicz	16/01/2023 - 23/05/2023
Inspiré de	Boris Vian	

PRESENTATION DU PROJET

- Le but de ce projet est de donner vie au pianocktail, instrument inventé par Boris Vian dans son roman *L'Écume des jours*. Il s'agit d'un piano à partir duquel on peut créer un cocktail en fonction du morceau joué. On y ajoute en plus une ambiance lumineuse associée.
- On joue sur 3 paramètres : l'intensité du son, la tonalité (note la plus présente dans le morceau), la hauteur (son grave ou aigu).

Schéma de principe



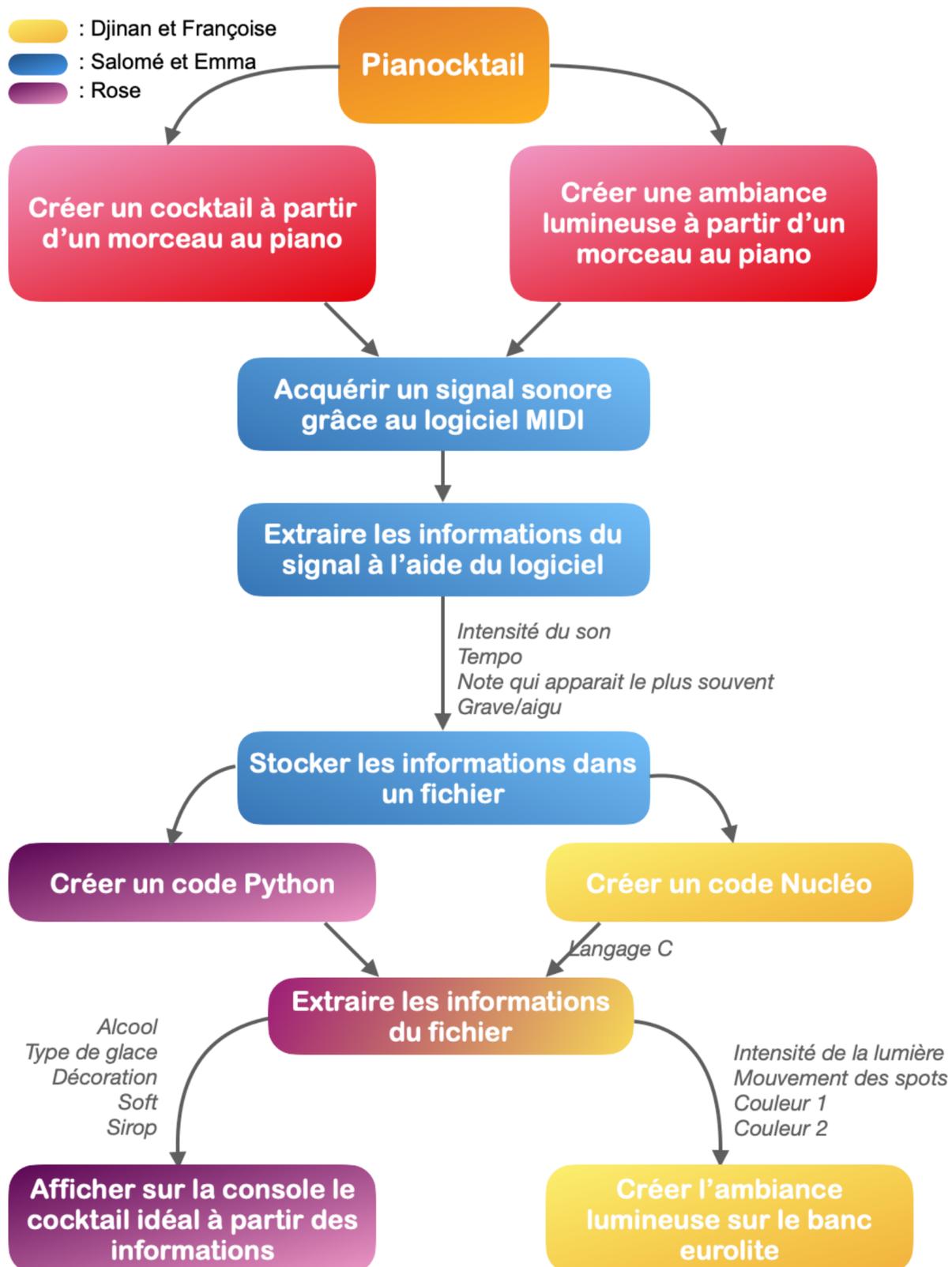
Notice d'utilisation

- L'utilisateur joue un morceau au piano relié à l'ordinateur. L'ordinateur affiche les paramètres recherchés sur sa console TeraTerm. L'utilisateur reçoit cette information et l'entre dans un code python qui lui dicte quel cocktail réaliser. Grâce au sirop utilisé, il peut choisir la lumière idéale à allumer.

Cahier des charges

Id	Exigence	Critère	Niveau	Flexibilité
1.1	Le pianocktail doit récupérer les bonnes informations musicales	Nombre de paramètres relevés nécessaire à l'identité d'une musique (Intensité du son, tempo, note qui apparait le plus souvent, son grave/aigu)	100% paramètres recherchés	±25%
1.2	Le pianocktail doit retranscrire les informations musicales en cocktail pour l'utilisateur	Nombre d'ingrédients contenus dans le cocktail (alcool fort, type de glace, soft, déco solide)	100% ingrédients recherchés	±0%
1.3	Le pianocktail doit retranscrire les informations musicales en ambiance lumineuse	Nombre de paramètres permettant de créer une ambiance lumineuse (mouvement des spots, deux couleurs différentes, intensité lumineuse)	100% paramètres recherchés	±0%

Schéma fonctionnel



Extraire les données du morceau de piano

- L'utilisateur joue sur un clavier relié au piano. Les notes jouées, ainsi que leur intensité, sont acquises sous forme de fichier MIDI (screen de MIDI Monitor). A chaque note correspond une lettre et un chiffre (correspond à un int sur C), et l'intensité est décrite par un chiffre. Les données sont directement stockées dans un tableau `tabvaleurs` et traitées par un programme rédigé sur MBed en langage
- Les notes et l'intensité sont classées entre 0 et 127. Les notes sont exprimées avec un nombre et une lettre, qui sont convertis directement en *int* par le programme.

Acquisition des données :

- `Tabvaleurs` est un tableau qui stocke la note jouée suivie de l'intensité avec laquelle elle est jouée. La dimension ne pouvant être fixée dynamiquement, on a choisi arbitrairement de ne conserver que les 500 premières notes jouées.
- Le code pour acquérir les données a été pris sur le LENSE.
- `Tabvaleurs2` contient uniquement les intensités et est construit en recopiant `tabvaleurs`

```
while(true) {
    /* MIDI */

    if(new_note_midi == 1){

        if(g < 1000){

            tabvaleurs[g]=note_data;
            tabvaleurs[g+1]=velocity_data ;
            g=g+2;

        }
        new_note_midi = 0;
    }
    ..... int tabvaleurs2[500];
    for (h=0; h<10; h++){
        tabvaleurs2[h] = tabvaleurs[k];
        k +=2;
    }
}
```

```
k = 0;
if(new_note_midi == 0 && g == 1000){

    for (k=0;k<1000;k++){//on cree un tableau ou l'on repertorie toutes les notes et l'on met à jour le no
        if (estdans(tabnotes,dim_tabnotes,tabvaleurs2[k])== 0 ) { //La note n'est pas dans le tableau
            tabnotes[i]=tabvaleurs2[k];
            tabnotes[i+1]=1;
            i=i+2;
        }
        else{// La note est dans le tableau
            j=indice(tabnotes, dim_tabnotes, tabvaleurs2[k]);
            tabnotes[j+1] = tabnotes[j+1] + 1;
        }
    }
    for (k=0;k<dim_tabnotes;k++){
        if (tabnotes[k] > 257){
            tabnotes[k] = 0;
        }
    }
}
```

- `Tabnotes` est un tableau qui stocke la liste des notes jouées suivies de leur occurrence. Il est construit dans la première partie du main, à l'aide des fonctions `est_dans` et `indice`.

Fonction *estdans()* :

- La fonction parcourt *tabnotes* pour savoir si la note *tabvaleurs[k]* a déjà été stockée. Si oui, on augmente de 1 le nombre d'occurrences grâce à *indice()*, sinon on ajoute la note dans le tableau.

- Fonction *indice()* :

- La fonction parcourt *tabnotes* pour trouver l'indice de *tabvaleurs[k]* dans celui-ci.

```
int estdans(int tab[], int dim, int n){
    int k;
    int nbr;
    for (k=0;k<dim;k++){
        if (tab[k]==n)
            return 1;}
    return 0;
}

int indice(int tab[], int dim, int n){
    //n la note que l'on cherche
    int k; //nombre d'occurrences
    for (k=0;k<dim;k++){
        if (tab[k]==n)
            return k;
    }
    return 0;
}
```

Traitement des données :

- On cherche à connaître le fondamental , l'intensité moyenne du morceau et la hauteur.

Calcul du fondamental

- On parcourt les nombre d'occurrences dans *tabnotes* pour trouver la note qui revient le plus

```
int h;
int ind=1;
int max=tabnotes[h+1];
for (h=0;h<dim_tabnotes;h+=2){
    if (tabnotes[h+1]>max){
        max=tabnotes[h+1];
        ind=h+1; // VALEUR DU FONDAMENTAL
    }
}
```

Calcul hauteur

- On récupère la moyenne des notes jouées grâce à *intensitemoyenne()*, et l'on considère ça comme le fondamental. On commence à parcourir *tabvaleurs* à l'indice 0, par pas de 2, pour accéder uniquement à la note.

```
int hauteurm;
hauteurm = intensitemoyenne(tabvaleurs, dim_tabvaleurs,0);

sprintf(debug_char, "hauteur = %d \r\n", hauteurm);
sPc.write(debug_char, strlen(debug_char));
```

Calcul de l'intensité

- On récupère la moyenne des intensités grâce à *intensitemoyenne()* . On commence à parcourir *tabvaleurs* à l'indice 1, par pas de 2, pour accéder à l'intensité.

```
int intensitem;
intensitem = intensitemoyenne(tabvaleurs, dim_tabvaleurs,1);
```

Le programme affiche ensuite le fondamental, l'intensité et la hauteur afin que l'utilisateur les entre sur l'interface régissant la lumière et le cocktail.

```
sprintf(debug_char, "intensité = %d \r\n", intensitem);
sPc.write(debug_char, strlen(debug_char));

sprintf(debug_char, "fondamental = %d\r\n", ind);
sPc.write(debug_char, strlen(debug_char));
```

Création du cocktail

- Pour la création du cocktail, on utilise un code python. Ce code utilise les données du piano pour trouver la composition parfaite du cocktail. Les notes et l'intensité sont classées entre 0 et 127. Les notes sont exprimées avec un nombre et une lettre, qui sont convertis directement en *int* par le programme.

Choix des ingrédients :

Description des fonctions

- Le choix de l'alcool composant le cocktail se fait en fonction de l'intensité du morceau. La fonction *choix_alcool* rend en entrée la valeur de l'intensité et renvoie l'alcool correspondant à cette valeur d'intensité.

```
def choix_alcool(intensite):  
  
    alcool='intensité inutilisable'  
  
    if ((intensite >= 1) and (intensite <= 18)) :  
        alcool='whisky'  
    elif ((intensite > 18) and (intensite <= 36)):  
        alcool='vodka'  
    elif ((intensite > 36) and (intensite <= 54)):  
        alcool='gin'  
    elif ((intensite > 54) and (intensite <= 72)):  
        alcool='tequila'  
    elif ((intensite > 72) and (intensite <= 90)):  
        alcool='manzana'  
    elif ((intensite > 90) and (intensite <= 108)):  
        alcool='malibu'  
    elif ((intensite > 108) and (intensite <= 127)):  
        alcool='martini'  
  
    return alcool
```

Pour les autres informations sur le cocktail à créer, on utilise les fonctions :

- *choix_soft* qui prend en argument la hauteur du morceau et renvoie le soft correspondant,
- *choix_sirop* qui prend en argument la note la plus utilisée qui renvoie quel sirop choisir,
- *choix_glace* qui prend en argument l'alcool utilisé et qui renvoie le type de glace adapté à l'alcool utilisé,
- *choix_deco* qui prend en argument la note la plus utilisée et qui renvoie la déco nécessaire pour que le cocktail soit sublimé.

Ces fonctions fonctionnent de la même manière que *choix_alcool* à l'aide de conditions *if/elif*.

Création du cocktail

- Les données du piano récoltées précédemment sont entrées manuellement après exécution du programme à l'aide de la commande *input()*.
- On stocke ensuite dans des variables les ingrédients correspondant aux données du morceau joué

```
#On entre manuellement les valeurs calculées par l'ordinateur  
alcool= int(input('intensité du morceau :'))  
  
hauteur = int(input('hauteur du morceau : '))  
  
deco = int(input('note la plus jouée : '))  
  
#On attribut aux variables son rôle dans le cocktail grâce aux fonctions écrites plus haut  
alcool = choix_alcool(alcool)  
deco = choix_deco(deco)  
soft = choix_soft(hauteur)  
sirop = choix_sirop(hauteur)  
glace = choix_glace(alcool)
```

Affichage du cocktail :

- Une fenêtre qui nous décrit le cocktail à réaliser s'ouvre et il n'y a plus qu'à réaliser notre cocktail. On utilise pour cela *PyQt* et on définit la classe *MainWindow* comme suit

```
# Affichage de la fenêtre avec le message pour décrire le cocktail
class MainWindow(QMainWindow):

    def __init__(self):
        super().__init__(parent=None)
        ''' Main Window parameters '''
        self.setWindowTitle("Votre cocktail idéal")
        self.setGeometry(100, 100, 500, 200)
        self.mainWidget = QWidget()
        self.setCentralWidget(self.mainWidget)

        ''' Graphical Elements '''
        self.my_label = QLabel(f'Ce morceau est incroyable ! \n Le cocktail parfait pour ce son a une base de {alcool}')
        ''' Layout Manager '''
        self.layout = QVBoxLayout()
        self.layout.addWidget(self.my_label)
        self.mainWidget.setLayout(self.layout)

    def closeEvent(self, event):
        QApplication.quit()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    main = MainWindow()
    main.show()
    sys.exit(app.exec_())
```

Mise en ambiance

- On choisit de réaliser une ambiance lumineuse. En effet, on pilote nos projecteurs lumineux reliés à l'ordinateur par une carte nucléo, à l'aide d'un code. On utilise en entrée le sirop/jus et le type de glaçon contenu dans le cocktail. Dans notre cas, les données sont entrées manuellement (dans la variable *data*). On réalise écrit plusieurs boucle if, qui pour chaque donnée d'entrée, contiennent le code qui permettent d'afficher la couleur désirée.
- Ici, le canal *dmx_data[3]* permet d'afficher une couleur bleue.

```
case GLACON:
    dmx_data[0] = 240;
    dmx_data[3] = 240;

    break;
```

BILAN

- Nous avons atteint notre objectif de produire un cocktail depuis un morceau joué. Cependant, nous n'avons pas réussi à coder l'acquisition du tempo du morceau, associé au sirop ajouté. Nous avons donc exploité la note moyenne la plus présente en changeant les intervalles car cette caractéristique était déjà associée à la décoration solide. De plus, on ne peut pas jouer un morceau de taille illimité avec notre pianocktail.
- Les problèmes de code et de sa syntaxe ont été les freins principaux à l'avancement de notre projet, mais le planning était dans l'ensemble respecté.
- Au niveau de l'exploitation d'un fichier où les paramètres devaient être enregistrés, nous n'avons pas pu le faire tout simplement car nous ne pouvons pas créer de fichier à partir de Nucléo, ce que nous ne savions pas. Nous avons pallié ce problème en entrant manuellement les données sur le code python.
- On s'est limité à faire seulement une ambiance lumineuse, sans jouer sur les mouvements des projecteurs par manque de temps.
- En ce qui concerne le travail d'équipe, les tâches ont globalement été bien réparties et le groupe présentait une bonne cohésion.

BILAN PERSONNEL

- Emma : Je me suis occupée avec Salomé de la partie «Traitement des données». Nous avons écrit un premier programme, et avons passé la majeure partie de notre temps à le débbugger. Françoise a finalement réussi à effectuer cette lourde tâche !
- Françoise : J'ai beaucoup aimé faire ce projet avec Rose, Djinan Emma et Salomé car il était très créatif. Le fait que Rose ait proposé de s'inspirer d'une œuvre littéraire a permis d'allier des compétences techniques à celles plus "libres". J'ai pu donc à la fois débbuggué un programme informatique, mais aussi réalisé deux mini-films pour présenter notre système. En guise d'intermédiaire, j'ai pu enfin transcrire les paramètres du cocktail en langage C !
- Rose : *L'écume des jours* fait partie des mes livres préférés et Boris Vian est une personne que j'admire. Donner vie au pianocktail est un objectif que j'ai depuis longtemps donc j'ai été honorée de participer à ce projet. Il n'est pas totalement aboutit mais je suis fière du travail qu'on a fournit.
- Djinan : le côté créatif du projet était très motivant et le travail en groupe était agréable . En m'occupant de l'ambiance lumineuse, j'ai appris à mieux gérer un code nucléo.
- Salomé : J'ai trouvé cela très gratifiant de parvenir à produire quelque chose proche de ce que l'on avait conceptualisé. La partie code nous a en effet pris la majeure partie de notre temps avec Emma, et je trouve cela peut être un peu frustrant de ne pas pouvoir participer à toutes les tâches pour appréhender le projet dans sa totalité.