Manuel Technique

Travail Pratique Individuel
MiFiSy

CFPT Informatique

Yoann Meier

6 mai 2024

Table des matières

Introduction

Ce document est un rapport présentant la conception du projet MiFiSy (MIDI Firework Symphony).
Ce projet est réalisé dans le cadre du projet de fin de formation au CFPT Informatique dans la formation *Développement d'applications*.
Il s'agit du *Travail Pratique individuel* (TPI).

MiFiSy est un projet monogame permettant de créer des feu d'artifice virtuel à l'aide d'une guitare MIDI.
Une musique de fond peut être ajoutée.
La séquence d'effet de feu d'artifice peut être sauvegardée au format XML et rejoué dans l'application.

Résumé du cahier des charges

Description de l'application

- Page d'accueil
  - Liste des musiques disponibles
  - Liste des séquences sauvegardées
  - Bouton pour aller dans le mode libre (musique optionnel)
  - Bouton pour aller dans le mode replay (choix d'une séquence obligatoire)

- Page de jeu, mode libre
  - 5 mortiers sont disposées uniformément au bas de l'écran avec un angle entre -10 et 10 degrés
  - Bouton de retour à l'accueil
  - Bouton de sauvegarde de la séquence de feu d'artifice au format XML
  - Jouer une corde déclenche un effet de feu d'artifice, la vélocité influe sur la vitesse ou la taille du feu d'artifice
    - Effet de comète :
      Une traînée lumineuse propulsée par un des 5 mortier choisis aléatoirement.
      La direction de la traînée est en fonction de l'angle du mortier.

    - Effet de pluie de particules :
      Des particules sont générés sur un point en haut de l'écran, celle-ci tombent avec l'effet de la gravité et disparaissent après un certain temps ou lorsqu'elles atteignent le bas de l'écran.

- Page de jeu, mode replay
  - La séquence joué choisis dans l'accueil est jouée
  - Les mortiers possèdent le même angle que dans la séquence
  - Bouton de retour à l'accueil
  - Les informations de la séquence (auteur, date, nom de la séquence) sont affichées

Element bonus

- Feu d'artifice
  - Étoiles simples (ou points)
    Un point lumineux qui apparait et disparait.

  - Feux d'artifice à explosion simple (pivoine)
    Une explosion basique qui se propage de manière uniforme dans toutes les directions.

  - Fontaines (pot au feu)
    Des particules jaillissent vers le haut avant de retomber, comme une fontaine.

Framework / librairie

Dans ce projet, j'utilise le framework Monogame pour développer mon application.
Monogame est un framework CSharp permettant de faire des jeux.

J'utilise également la librairie NAudio permettant de récupérer les notes joué par une guitare MIDI en format MIDI.

Outils / logiciels utilisés

- Ordinateur Windows 10
- CSharp - Visual Studio 2022
- Framework Monogame
- GitHub
- Google Drive
- LaTeX
- Suite office

Livrables

- Manuel technique
- Manuel utilisateur
- Journal de bord
- Rapport TPI
- Le projet

Méthodologie

Méthodes en 6 étapes

Pour assurer le bon déroulement de mon projet, j'ai utilisé une méthodologie de travail afin d'être organisé et efficace.
Après avoir examiné différentes méthodes lors de ma formation, j'ai opté pour la méthode en 6 étapes.
Je l'ai choisie car, parmi les différentes méthodes étudiées, elle s'est révélée être la plus adaptée pour un travail individuel à court terme.

- **S'informer**
  Au début du projet, j'ai analysé le cahier des charges afin de comprendre toutes les fonctionnalités à réaliser.
  J'ai également demandé des précisions à mon formateur sur certains points que je n'ai pas compris.
- **Planifier**
  Une fois le cahier des charges compris, j'ai préparé le planning de mon travail en découpant le travail par tâches avec une durée prévisionnelle.
  Voir : Planning prévisionnel
- **Décider**
  Dans la phase de décision, j'ai décidé l'ordre de réalisation de toutes les tâches en fonction de l'importance de celle-ci sur le projet.
- **Réaliser**
  Cette étape est cruciale dans mon projet, car elle implique la concrétisation de mon travail en suivant la planification établie précédemment. Grâce à une bonne compréhension des tâches à accomplir, j'ai pu effectuer mon travail dans des conditions optimales.
  Voir : Code Source
- **Contrôler**
  Cette étape est importante car elle consiste à contrôler le bon avancement du projet en s'assurant que les objectifs du travail sont atteints. A chaque tâche terminée, j'ai effectué des tests pour vérifier le bon fonctionnement de ces tâches.
  Voir : Rapport de test
- **Évaluer**
  Pour finir, j'ai évalué tous les résultats obtenus et j'ai également réfléchi aux différents moyens d'améliorer mon travail, que ce soit pour ajouter de nouvelles fonctionnalités ou améliorer mon code.
  Voir : Amélioration possible

Analyse fonctionnelle

Vue Accueil

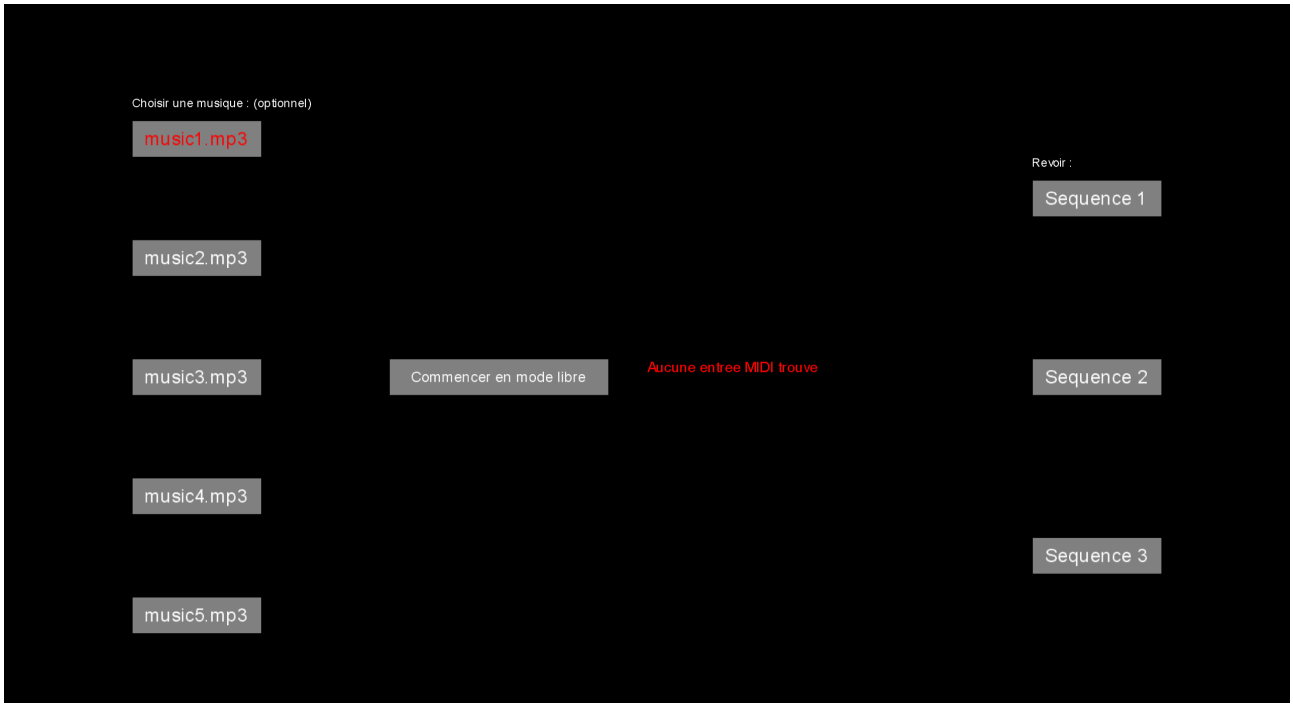

FIGURE 1 – page de démarrage de l'application

Dans cette page, un message d'erreur apparaît au milieu de l'écran si aucune entrée MIDI n'est détectée : "***Aucune entrée MIDI trouvé***".

Si l'utilisateur souhaite une musique de fond pendant le mode libre, il peut appuyer sur l'une des musiques affichés à gauche. Chaque musique provient d'un dossier qui peut être défini dans le fichier de configuration. Lorsqu'une musique est sélectionne, sa couleur deviendra rouge (music1.mp3 sur l'exemple).

L'utilisateur peut appuyée sur le bouton "***Commencer en mode libre***", ce qui le redirigera vers la page de jeu en mode libre.

A droite de l'écran, les différentes séquences précédemment enregistrer sont affichés.
Lors d'un clique sur l'une d'entre elle, l'utilisateur sera redirigé sur la page de jeu en mode replay.

Vue Jeu

Mode libre

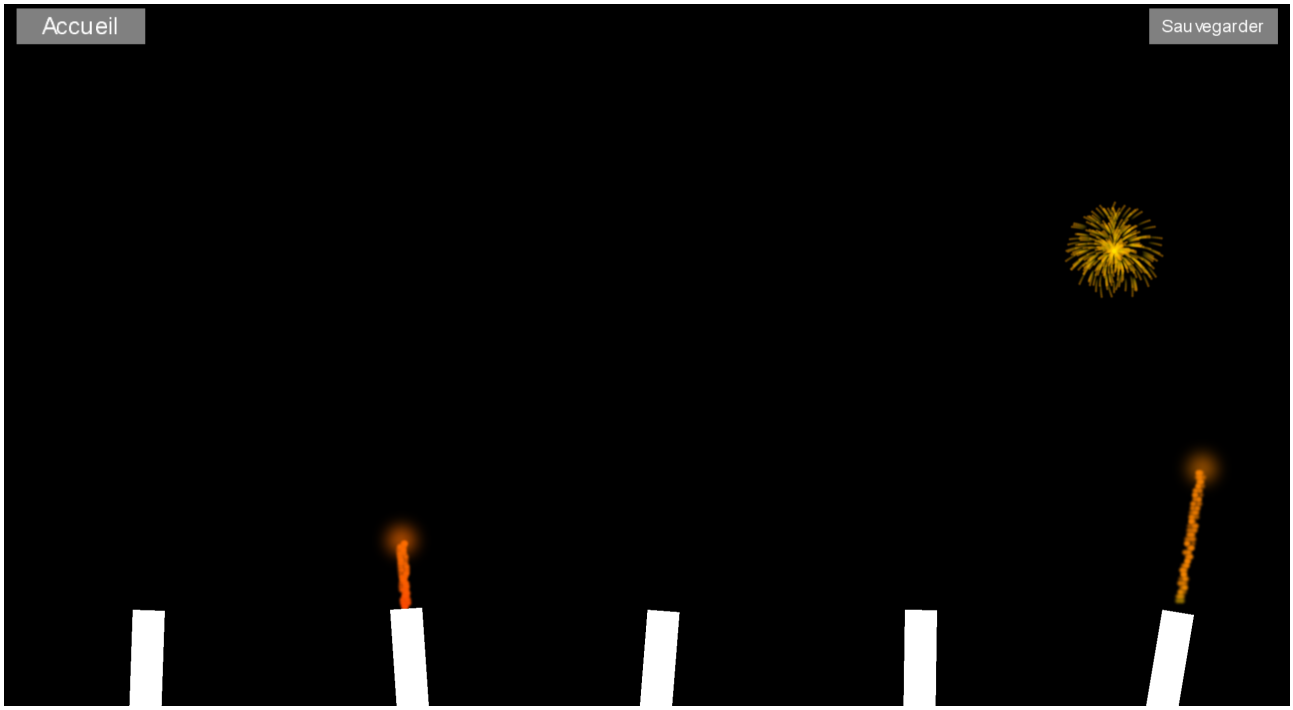

FIGURE 2 – Mode de jeu libre

Dans cette page, un bouton "***Accueil***" permet de retourner à l'accueil, ce qui remet à 0 la séquence de feu d'artifice.

Plusieurs mortiers, défini dans le fichier de configuration (position x et y, taille en hauteur et en largeur et l'angle d'inclinaison) sont affichés.
Si aucun mortier n'est défini dans le fichier de configuration, 5 mortiers sont disposées uniformément en bas de l'écran avec la même taille et un angle entre -10 et 10 degrés.

Si l'utilisateur (connecté à une guitare MIDI) joue la première corde (tout en bas), la comète sera lancer aléatoirement d'un des mortier représenté par un rectangle blanc avec la même direction.
La vitesse de déplacement dépend de la vélocité à laquelle la corde est jouée.

Si l'utilisateur joue la deuxième corde, la pluie de particule est crée sur un point aléatoire du haut de l'écran.
La durée de vie du feu d'artifice dépend de la vélocité à laquelle la corde est jouée.

Le bouton en haut à droite permet de sauvegarder la séquence de feu d'artifice crée au format XML.
Un message de confirmation de sauvegarde apparaît brièvement sur l'écran.

Mode replay

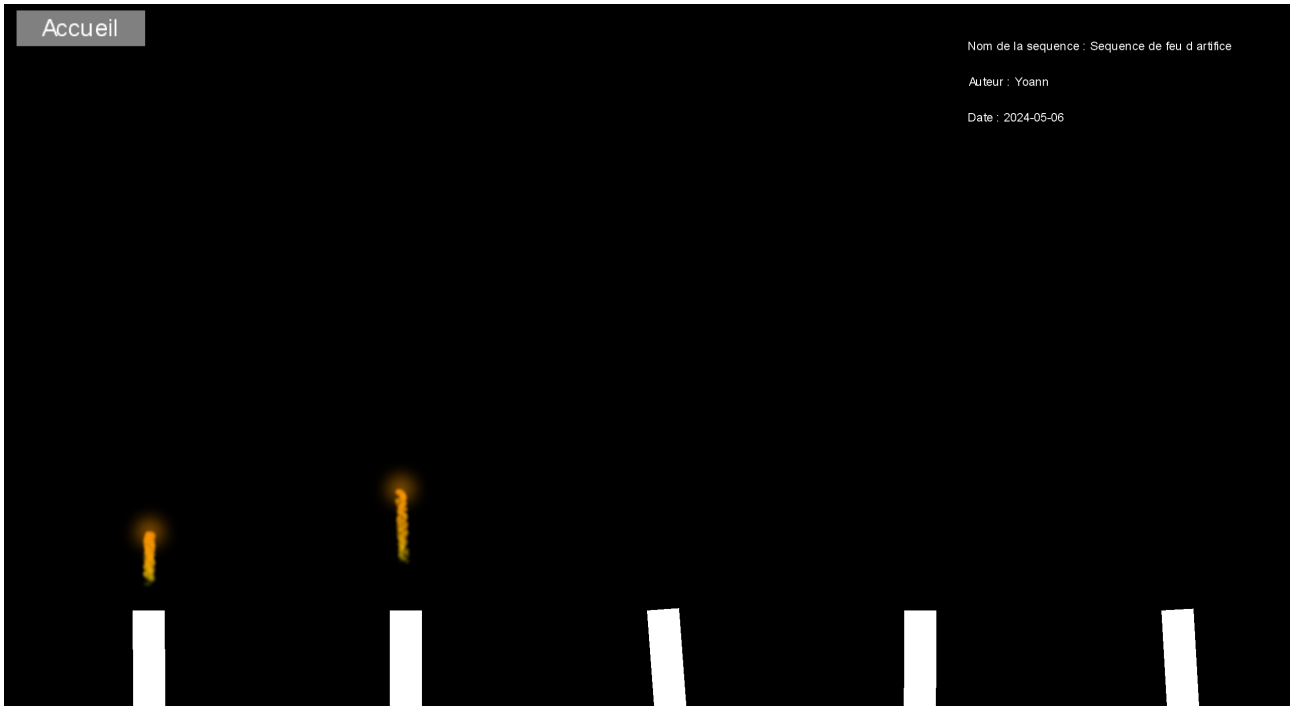

FIGURE 3 – Mode de jeu replay

Dans cette page, une séquence sauvegardé est jouée, la musique, l'image de fond et la position des mortiers est identique, les informations de la séquence sont écrits en haut de l'écran.
A la fin de la séquence, un message ("***Sauvegarde effectue***") apparaît pour notifier la fin de l'enregistrement à l'utilisateur.
Le bouton "***Accueil***" permet de revenir à l'accueil.

Analyse organique

Structure des fichiers

Diagramme de classe

Plan de test et tests

Périmètre de test

Pour MiFiSy, je vais créer un plan de test visant à garantir le bon fonctionnement de l'application du point de vue de l'utilisateur.
Ce plan inclura des tests fonctionnels pour évaluer à la fois la performance et la convivialité de l'interface utilisateur.
Ensuite, je consignerai tous les tests réalisés ainsi que leurs résultats dans un tableau pour assurer la qualité de l'application et suivre l'évolution du projet.

Plan de test

N°	Description du test	Résultat attendu
1	Lors d'un clique sur une musique d'ambiance	La couleur du nom de la musique change et la musique est sélectionnée
2	Lors d'un clique sur une musique d'ambiance puis sur le mode libre dans l'accueil	La musique sélectionnée est jouée en boucle dans le mode libre
3	Lors d'un clique sur le mode libre dans l'accueil sans cliquer sur une musique	Le mode libre se lance sans musique
4	Lors d'un clique sur un replay dans l'accueil	Le replay se lance, la musique et les effets sont identiques
5	Lorsque le replay est terminée	Un message l'indiquant apparaît à l'écran
6	Lors du clique sur 'Accueil' dans le mode libre ou replay	Retour à la page d'accueil
7	Lors du clique sur 'Sauvegarder' dans le mode libre	Toute la séquence crée est sauvegardé dans un fichier xml et un message "Sauvegarde réussi" apparaît brièvement au milieu de l'écran
8	Dans le mode libre, lorsque la première corde de la guitare est joué	L'effet de comète est crée sur un des mortiers aléatoirement
9	Dans le mode libre, lorsque la deuxième corde de la guitare est joué	L'effet de pluie de particules est crée aléatoirement sur le haut de l'écran
10	Si la guitare n'est pas trouvé lors du démarrage de l'application	Un message d'erreur apparait à l'écran

Evolution des tests

N Test	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11
1	✗	✗	✗	✗	✓	✓					
2	✗	✗	✗	✗	✓	✓					
3	✗	✗	✗	✗	✓	✓					
4	✗	✗	✗	✗	✗	✓					
5	✗	✗	✗	✗	✗	✓					
6	✗	✓	✓	✓	✓	✓					
7	✗	✗	✗	✓	✓	✓					
8	✗	✗	✗	✓	✓	✓					
9	✗	✗	✗	✓	✓	✓					
10	✗	✗	✗	✗	✓	✓					

Rapport de test

N°	Date du test	Résultat obtenu	OK/KO

Conclusion

Difficultés rencontrées

Amélioration possible

Bilan personnel

Annexes

Planning prévisionnel

Planning effectif

Code Source

```
15              var mouseState = Mouse.GetState();
16
17              HasClicked = mouseState.LeftButton == ButtonState.Pressed && ↩
                    _lastMouseState.LeftButton == ButtonState.Released;
18              MousePosition = mouseState.Position.ToVector2();
19
20              _lastMouseState = mouseState;
21
22              //
```