

P1611978 LAFOIX Paul
P1611312 POLOCE Antoine
P1307400 RABUT Théo
P1410782 LANGLADE Maxime

Cahier Des Charges

Nom du projet

I- Presentation

Dans le cadre de l'UE LIFAP4 Conception et développement d'application nous avons choisi de développer un jeu. Avec les contraintes données et nos goûts nous avons choisis de se rapprocher d'une application musicale et ludique.## Notre jeu sera composé d'un affichage graphique d'une partition synchronisé avec la lecture audio, d'une fenêtre de validation, mais aussi d'un mode création de partition. Beaucoup de jeux de ce type ont été fait, avec des fonctionnalités ressemblantes à celles de notre projet, mais nous allons être original et rendre <NOM DU PROJET> unique.

Règles du jeux :

Sur votre écran, des « notes » vont défiler et vont avoir une certaine synchronisation avec le fichier audio que vous aurez préalablement choisi.

Le but du jeux étant de taper sur la touche de votre clavier au moment ou la « note » arrivera dans la zone de validation. Et ainsi d'exploser tout les records.

Vous trouverez le fichier Doc/README.txt dans l'archive du jeux. Ce fichier texte vous expliquera explicitement les règles du jeu et la marche à suivre pour que l'expérience de jeu se retrouve optimale.

II- Description de la demande

Au terme de cette UE nous voulons avoir un jeu cohérent, ludique et original. Il n'y aurait pas de sens à faire une application de simulation d'un instrument qui serait trop simple et sans difficulté nous allons donc avoir un rendu final amusant.

Pour cela, nous devons avoir :

- Un menu qui permet de choisir le mode de jeux (Création/Jouer), les paramètres, la chanson, la difficulté...
- Un affichage graphique dynamique des partitions, les « notes » vont devoir arriver de loin et venir sur la barre de validation.
- Une gestion de score, et de par ce fait une gestion de la validation des notes au cours d'une partie.
- Un mode création, qui permettra de créer des partitions qui seront compréhensibles par nos algorithmes de jeux.
- Un système de partition qui nous permettrait de lire un fichier audio et le transformer en un fichier lisible par un algorithme simple. (fichier type txt)
- Jouabilité via clavier.
- Si possible, jouabilité avec guitare électronique (type GuitarHero).
- Plusieurs niveaux de difficultés pour un seul fichier audio :

Ex : Dragon Force - through the fire and flame

Niveau Expert, Intermédiaire, Débutant.

Le cahier des charges étant prévisionnel, nous allons sans aucun doute modifier, enlever et rajouter des fonctionnalités dans notre application.

III- Contraintes

Pour réaliser notre application nous devons suivre une méthode de conception qui nous est fourni : la méthode AGILE.

Le langage utilisé est du C++, le C n'est pas à proscrire mais nous devons tout faire pour nous en passer.

La librairie utilisée pour les graphismes sera SDL2.

Nous devons fournir une documentation claire et compréhensible, pour cela nous allons utiliser Doxygen, cette documentation devra être mise à jour régulièrement.

Nous utilisons un outil de gestion de version : Mercurial.

La conception d'une telle application nécessite une organisation très précise et une compréhension parfaite de la phase de conception pour cela nous allons utiliser un diagramme des classes qui éclaircira chaque partie non triviale.

Le code sera fourni à la société InfoGamesProf.

Encore une fois, les fonctionnalités de <NOM DU PROJET> vont évoluer, on ne peut pas vraiment prévoir quelle seront les librairies que nous allons utiliser.

IV- Déroulement du projet

0 Version non jouable (Terminal)

- 0 Analyse de la version minimale
 - 0.1 Les classes indispensables et le coeur du jeux
 - 0.2 Le format des fichiers de lecture/ecriture
 - 0.4 UML et premier diagramme de Gantt
- 1 Conception architecturale et premiers Tests
 - 1.0 .h généraux à faire avec priorité sur le coeur du jeux défini en 0
 - 1.1 Affichage en mode console des notes et des temps.
 - 1.2 .h plus spécifique et type de données efficaces
- 2 Conception détaillé, début de code et Tests
 - 2.0 Implémenter le corps du jeu.
 - 2.1 Test et vérification des besoins
- 3 Fin de code et upgrade
 - 3.0 Finition du code et test d'utilisation.
 - 3.1 Si validation, on passe à la version avec affichage graphique et ludique. (SDL2)

L'ordre des tests et de l'implémentation de nouvelles fonctionnalités est difficile à prévoir, mais nous avons ici le minimum à faire pour avoir un programme fonctionnel, ce qui est notre but dans un premier temps.

Dans un deuxième temps nous allons nous pencher plus en détail sur l'affichage graphique :

- 0 Recherche SDL2 en fonction des besoins.
- 1 Implémentation des modifications de notre code existant.
- 2 Debug, Fin de code et Test d'utilisation.

Certaines implémentation ne mérite pas d'être traitées et faites en priorité, puisqu'elle ne sont pas indispensables au bon fonctionnement du jeux. Elles le seront dans un second temps pour le coté ludique. (ex : la fluidité des notes, la transformation d'un fichier .wav en un fichier lisible par notre code, la qualité des animations et des images,...)

Il n'y a pas (pour le moment) d'attributions de taches explicites, nous sommes une équipe qui communiquera, qui partagera son travail et qui avancera en tant que tel. Si la répartition du travail venait à être trop inégales, nous aviserons en conséquences.