

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 dynamique d'une partition, la musique qui tournera en audio et sera synchronisé avec l'affichage graphique et d'une fenêtre de validation pour chaque notes. 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.

##Plus de règles dans le README, il est fait pour ça.

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 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)
- 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.
- ##Gestion de la réussite.
- Un mode création, qui permettra de créer des partitions qui seront compréhensibles par nos algorithme de jeux.
- 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.

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 propre et ludique. (SDL2)

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

1 Version Jouable (SDL2)

- 0 Recherche librairies SDL2, openGL... selon les besoins.
 - 0.0 Recherche sur la librairies SDL (nous suffira t-elle?)
 - 0.1 Declaration des fonctions d’affichage dans les .h et creation des nouveaux modules si besoin est.
- 1 Implementation de la version Affichage propre
 - 1.0 Code des .cpp correspondants au .h créés ou modifiés en 1.0.1
 - 1.1 Debug et test d’affichage basique
 - 1.2 Verifications de la cohérence audio/video
- 2 Test d’utilisation et validations
 - Test d’utilisation, en lien avec 1.1.2
 - Si validation, on passe en 2.0.0 avec une analyse des besoins orientés vers la jouabilité et le game design (le cycle pour implementer de nouvelles fonctionnalités ressemblera surement au cycle 0)
- Recherche SDL2,
- Implémentation des modifications de notre code existant.
- Debug, Fin de code et Test d’utilisation.

Il n’y a pas d’attributions de taches explicites, nous sommes une équipe qui communiquera, qui partagera son travail et qui avancera en tant que tel. Si la repartition du travail venait à être trop inégales, nous vous le ferons savoir lors de notre soutenance.