

Spécification

Projet DeViNT

1. Persona choisi

Identité : Liam

Données démographiques : Adulte de 30 ans

Activité professionnelle : Travail à la chaîne

Activités domestiques et de loisirs : Ecoute de la musique, adore les quizz musicaux, pratique le sport et a un esprit créatif. Il aime fabriquer des choses.

Handicap : Déficience visuelle partielle. Troubles cognitifs associés.

Buts et tâches : Être capable de réaliser des tâches de la vie quotidienne sans assistance. Il souhaiterait savoir jouer de la musique.

Usage des technologies : Sait se servir d'une souris.

Attitudes à l'égard des technologies : Apprécie les jeux sur ordinateur, garde une excellente expérience des jeux DeViNT de l'année précédente.

Communication : Introverti, sociable, ne passe pas beaucoup de temps devant l'ordinateur.

Suggestion : Quizz musicaux ou jeu ayant un lien avec le sport

Notre projet vise un public ne présentant aucun trouble de l'audition et appréciant la musique, il se veut accessible à toute personne capable d'utiliser une souris.

2. Sujet détaillé

Notre jeu va considérer en un échantillonneur 'sampler' de musique. L'utilisateur aura besoin d'avoir une audition suffisante pour apprécier le jeu. L'aspect visuel restera assez accessible : il consistera en des tuiles de différentes couleurs représentant les différents instruments. Ainsi en sélectionnant certaines tuiles, par exemple une pour les percussions, une autre pour l'accompagnement et une pour la mélodie le joueur pourra créer un son comme il le voudra. Bien sûr les sons seront enregistrés pour être compatibles et ils seront répétables à l'infini, pour une longueur d'entre 5 et 10 secondes par boucle.

Au début nous pensons faire environ 5 instruments chacun composés de 5 variantes, il sera possible par la suite d'étendre le nombre de possibilité, par exemple via une sélection dans un menu afin de ne pas surcharger l'écran.

Nous pensons que ce sujet est adapté au public visé voir même un peu plus car c'est quelque chose d'assez intuitif et qui pousse à l'expérimentation et à la création dans un domaine qui plaît à la cible, de plus c'est un divertissement assez basique qui permet des sessions très courtes voir de durée moyenne, sans pouvoir perdre ou gagner.

3. Scénario

L'utilisateur lance notre programme. Il arrive sur un écran d'accueil présentant le titre du jeu, et des éventuelles options. Il peut choisir soit de quitter ou d'accéder à l'écran de jeu principal.

Lors du lancement du jeu en lui même, le joueur se voit présenté l'écran de la figure suivante.



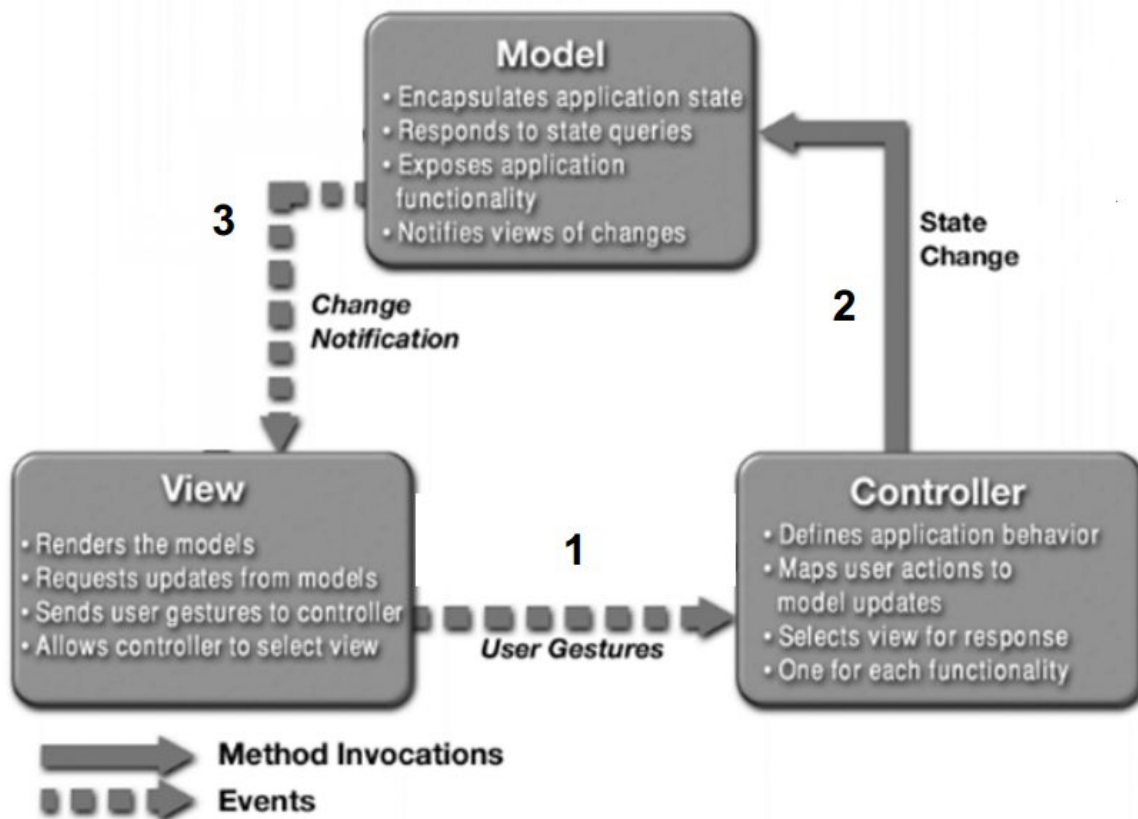
Le loop musical démarre lorsque l'on clique sur une des tuiles. A chaque tile correspond un échantillon musical particulier. Cliquer une fois sur une tile active un échantillon particulier, et la superpose aux autres échantillons actifs. Cliquer une seconde fois la désactive. Ces changements sont effectués à chaque fois qu'une nouvelle boucle musicale reprend.

Une barre de progression en haut de l'écran affiche l'avancement de la boucle musicale actuelle, et repart à zéro à chaque cycle musical.

Le jeu ne possède pas réellement de fin à proprement parler : l'utilisateur peut quitter à tout moment en utilisant la touche escape dédiée.

4. Modèle des classes

L'architecture la plus adaptée à notre concept de jeu est a priori le modèle MVC classique, la partie modèle sera constituée de notre banque de sons qui sera interrogée via la vue fxm1 et l'intervention du contrôleur pour décoder les signaux envoyés.



5. Planning de réalisation

Semaine 8 : Mise en place de l'architecture du projet + Prise en main de l'outil deViNT

Semaine 9 : Création des vues FXML + Rattachement à l'API deViNT

Semaine 10 : Terminer vues FXML et contrôleur basique + Commencer à enregistrer/récupérer des sons

Semaine 11 : Continuation de la semaine 11 + Mise en place de la logique algorithmique du contrôleur

Semaine 12 - 13 : Avoir un projet minimum jouable : un échantillon musical par instrument pouvant se superposer les uns les autres. Les échantillons se lancent dès que l'on clique dessus, sans attendre la fin des loops qui sont déjà en lecture.

Semaine 14 : Éliminer les bugs restant

+Lancement des échantillons musicaux synchronisés, quand les boucles de tous les autres extraits musicaux en lecture démarrent.

+ Ajout de nouveaux extraits musicaux

DEMONSTRATION

Semaine 15-16 : Prendre compte du feedback afin d'apporter des améliorations aux principes de jeu

+ Eliminer les bugs

+ Ajout d'une interface de sélection de set d'instruments

+ Ajout de nouveaux extraits musicaux

+ Commencer le système d'enregistrement des musiques composées

Semaine 17-18 : Terminer l'enregistrement des musiques composées

+ Option de modification des musiques composées.

Semaine 19-20 : Finir les améliorations et s'assurer que toutes les fonctionnalités marchent correctement

+ Ajout de nouveaux extraits musicaux

Semaine 20-22 : Régler tous les bugs potentiels

+ finaliser le rendu

DEMONSTRATION FINALE

-Les tests unitaires seront effectués en continu

-Les tâches seront distribuées au fur et à mesure

Tâches principales du projet :

-Enregistrer des samples audio

-Se rattacher à l'API deViNT

-Créer la vue et le design de l'application

-Coder la logique derrière le principe du jeu

- Lecture de fichiers audio

- Lancement synchronisé de fichiers audios

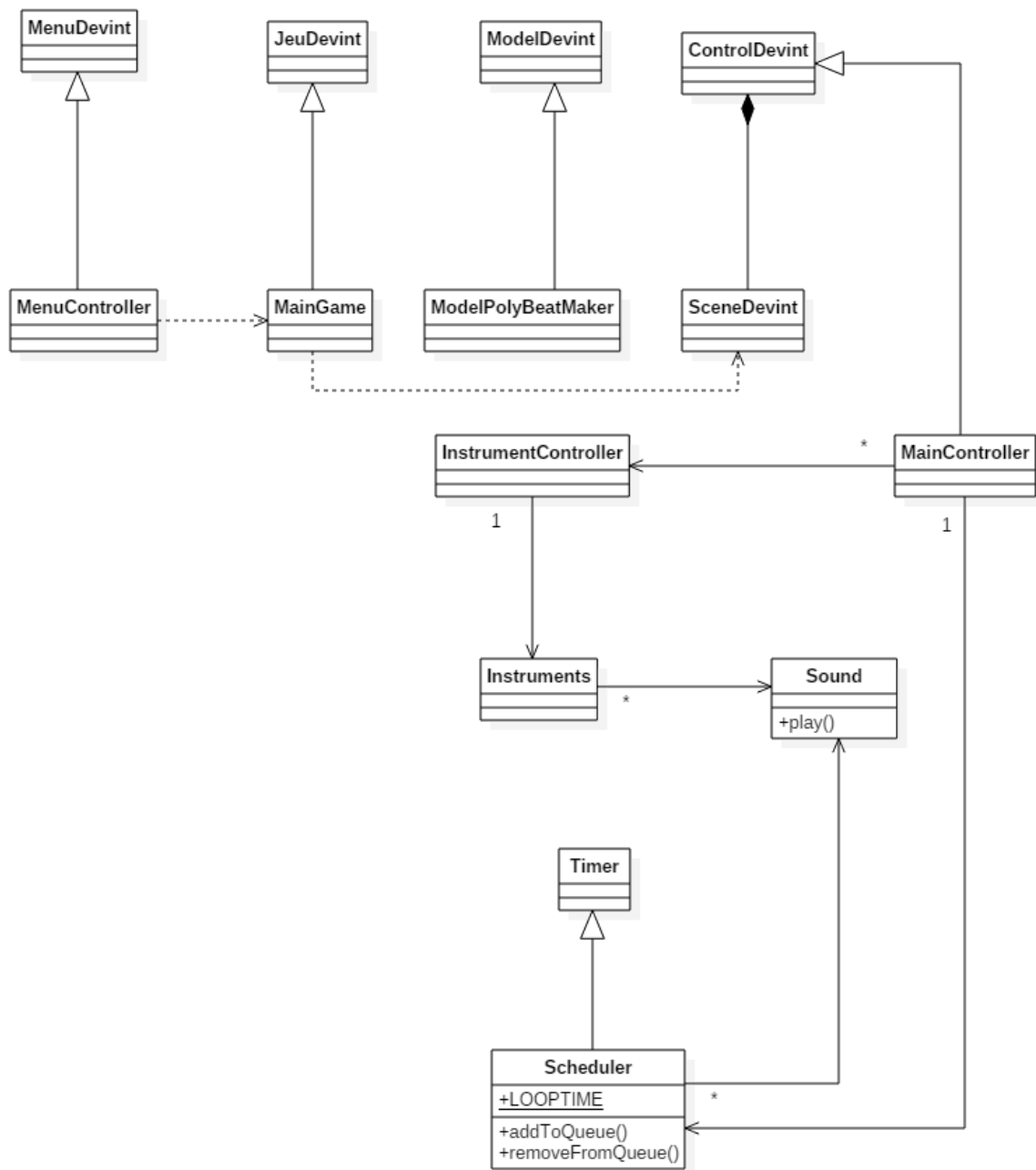
- Enregistrement de fichiers de préférences pour relire les compositions musicales

- Modifier des fichiers de préférences des compositions.

-Fonctionnalités avancées du projet

-Modifications inattendues (feedback client)

6. Diagramme de classe simplifié



MenuController implémentant l'interface de menu devint instancie notre jeu principal *MainGame*. Celui-ci crée à son tour le modèle de jeu devint ainsi que sa scène et son contrôleur *MainController*.

Une fois le contrôleur principal chargé, celui-ci charge ses composants FXML et leurs contrôleurs associés *InstrumentController* à partir des classes de l'énumération *Instruments*.

Chacun des instruments a ses sons associés. Le *MainController* se charge également du *Scheduler*, qui joue tous les sons actifs toutes les *LOOPTIME* secondes.

Lorsque un son est activé, le contrôleur de l'instrument le relaie au contrôleur principal, qui le transmet au *Scheduler* afin qu'il soit joué. Les sons sont retirés de la liste de lecture du *Scheduler* selon le même schéma.