

Mode d'emploi de la base pour les projets DeViNT

Ce document est un manuel d'utilisation de la base des projets DeViNT. Il est destiné aux élèves de 1^{ère} année, afin de les guider et de leur donner des consignes précises pour la création de leur projet.

Il contient les sections suivantes :

1. Présentation
2. Architecture de la base
3. Quelques conseils à appliquer pour coder un jeu destiné à des DV
4. Important : Avant de commencer à coder
5. Où coder votre jeu ?
6. Comment ajouter des touches pour votre jeu ?
7. Où coder les règles du jeu ?
8. Qu'y a-t'il à votre disposition ?

Il est conseillé de lire l'intégralité des consignes avant de commencer à coder, pour ne pas avoir à modifier des choses dans votre code par la suite.

1. Présentation

Pour créer votre projet DeViNT, vous allez partir d'une base qui vous est fournie. Cette base contient un menu principal contenant les 4 boutons suivants :

- * Jouer : lance le jeu (que vous allez créer)
- * Configuration profil : panneau de réglages de l'interface (choix couleurs, voix...)
- * Règles : panneau contenant les règles de votre jeu (à compléter)
- * Quitter : quitte l'application

Cette base est destinée à fournir une interface agréable et commune pour les déficients visuels. Vous DEVEZ vous en servir pour réaliser votre projet. Vous trouverez dans cette notice les consignes d'utilisation de cette base.

Remarque sur l'interface de configuration : Le sous-menu « Configuration profil » est destiné à permettre aux déficients visuels de configurer des profils adaptés à leur vision. Vous pouvez utiliser ce menu pour voir si votre jeu est adapté aux changements de couleurs, de voix...

2. Architecture de la base

Le projet possède l'architecture suivante :

- **classesMeres**
 - *CommandeInterface* : une interface contenant la liste des actions possibles
 - *ClasseAbstraite* : une classe mère qui implémente *CommandeInterface*
 - *MainFrame* : hérite de *JFrame* et de *KeyListener*. Elle permet de switcher entre les différents panels et d'affecter les actions de *CommandeInterface* aux touches.
 - *Launcher* : contient le main du projet
 - *Settings* : contient tous les attributs que vous devrez utiliser.
- **menuGraphique**
 - *MenuPanel* : le *JPanel* du menu
 - *InterfaceConfigurationProfil* : hérite de classe abstraite et contient le *configPanel*
 - *ConfigPanel* : le *JPanel* permettant la configuration
 - *ConfigurationProfil* : permet la gestion des profils avec les fichiers de profils
 - *ConfigurationSiVOX* : permet la gestion du fichier *SiVOX.conf*
- **jeu**
 - *InterfaceJeu* : hérite de classe abstraite et contient le *JeuPanel* et le *TouchePanel*
 - *JeuPanel* : le *JPanel* principal du jeu
 - *TouchePanel* : le *JPanel* de rappel des touches
- **regle**
 - *InterfaceRegle* : hérite de classe abstraite et contient *ReglePanel* et *TouchePanel*
 - *ReglePanel* : le *Jpanel* principal des regles

Les classes à modifier sont les suivantes :

- **classesMeres**
 - *CommandeInterface* : déclarer les nouvelles actions
 - *ClasseAbstraite* : ajouter le corps par défaut des nouvelles actions
 - *MainFrame* : affecter une touche à chaque action
- **jeu**
 - *InterfaceJeu* : implémenter les actions des touches
 - *JeuPanel* : implémenter l'interface graphique du jeu
- **Regle**
 - *InterfaceRegle* : modifier le texte du rappel des touches spécifiques
 - *ReglePanel* : implémenter l'interface graphique des règles

Les fichiers **compilation.bat** et **execution.bat** servent à **compiler les classes**, et **lancer l'application** :

- **Compilation.bat** se trouve dans le répertoire **src**.
- **Execution.bat** se trouve dans le répertoire **bin**.

Pour se servir du **VocalizeSiVOX**, vous devez placer ce dossier dans le même répertoire que celui dans lequel vous créez le répertoire de votre jeu. Ensuite il suffit d'importer le *SI_VOX.jar* contenu dans le sous-répertoire **bin**.

Diagramme de classes

3. Quelques conseils à appliquer pour coder un jeu destiné à des DV

Vous créez une application destinée à des déficients visuels. Vous ne devez pas oublier ceci pendant la durée du projet ! Votre but n'est pas de créer un jeu en 65 000 couleurs, mais de respecter la demande : un jeu utilisable par des déficients visuels.

Libre à vous d'améliorer votre jeu par la suite, dans un contexte personnel, pour proposer par exemple une version plus évoluée, destinée aux voyants.

Des consignes précises vous sont données dans la section 3 : « Important : avant de commencer à coder ». Cependant, en règle générale, vous devez déjà suivre les conseils suivants, issus des expériences des années précédentes :

3.1. A propos des touches

Certaines touches ont déjà été implémentées. Par exemple, dans le menu principal, vous naviguez avec haut et bas, validez avec entrée, etc. Vous découvrirez ces touches plus loin dans ce manuel.

Pour votre JEU, vous devez, suivre les conventions de raccourci et de commandes clavier suivantes :

- * F1 = Rappel des touches d'aide (F1, F2, F3, F4, F5).
- * F2 = Rappel des règles du jeu
- * F3 = Scores du jeu
- * F4 = Rappel des touches du jeu
- * F5 = Donner l'information courante
- * Enter = Valider une action
- * Ctrl = Afficher/Cacher le panneau des touches
- * Echap = Revenir au menu / Quitter

Pour la manière d'implémenter les touches, voir la section 6.

Ensuite, vous pouvez utiliser d'autres touches, selon les besoins spécifiques de votre jeu, pensez à utiliser au besoin :

- * La touche Espace
- * N'oubliez pas que les touches F et J sont facilement repérables au toucher.
- * Mieux vaut éviter d'utiliser le pavé numérique (les portables n'en n'ont pas).

3.2. Comment créer une application accessible ?

Si vous utilisez le clavier alphanumérique pour entrer un nom ou des commandes, il peut être utile :

- * de dire à voix haute les touches pressées,
- * de permettre de corriger au besoin avant de valider la saisie clavier.
- * d'ignorer la casse, d'accepter indifféremment majuscules et minuscules.

Vous devez informer l'utilisateur sur l'état et l'évolution du jeu, sans que cela devienne fastidieux pour lui. En utilisant de préférence la synthèse vocale.

Le joueur doit :

- * savoir que l'on débute le jeu ou qu'on le termine
- * savoir que l'on entre dans une nouvelle phase du jeu
- * savoir ce qu'on attend de lui (exemple : quand c'est à lui de jouer)
- * connaître les événements qui surviennent au cours du jeu
- * savoir que ses actions sont perçues par l'ordinateur (retours)
 - * les coups des adversaires humains ou logiciels
 - * le déroulement du jeu : tel gagne, perd, tel ramasse le tas...

Pour éviter la lourdeur de cette information sur le jeu, l'interface doit être très réactif, et l'utilisateur doit être sûr qu'il le contrôle en permanence, pour cela :

- * la synthèse vocale est interrompue dès qu'une touche est pressée (déjà implémenté)
- * F5 renvoie les informations courantes sur (à qui le tour, qui a quoi, où en est on...)
- * Avant de jouer un nouveau son, il faut s'assurer qu'on a coupé les sons précédents.

Puisque le jeu s'adresse aux aveugles et aux malvoyants, il faut privilégier l'information sonore mais il faut soigner aussi l'information visuelle (gros des éléments visuels).

La taille de la police sera choisie par l'utilisateur, mais vous devez utiliser du ARIAL.

C'est un plus de pouvoir jouer aussi bien avec les raccourcis clavier qu'avec les clics souris.

4. Important : Avant de commencer à coder...

Cette section contient les consignes précises que vous allez devoir suivre pour coder votre application.

La base qui vous est fournie possède des attributs de couleur, voix, taille de police. Ceci est destiné à ce que les futurs utilisateurs des jeux DeViNT puissent régler ces paramètres à leur convenance.

Vous disposez de :

- * 5 couleurs (pour : le fond, le texte, le joueur 1, le joueur 2, les objets).
- * 3 voix (pour : voix OFF, voix joueur 1, voix joueur 2)
- * une taille de police

VOUS DEVEZ UTILISER CES ATTRIBUTS

Vous n'êtes pas tenus de les utiliser tous, mais vous devez au moins en utiliser certains (votre jeu a sans doute un fond, et des éléments, et vous devriez avoir besoin d'au moins une voix !).

4.1. Pourquoi utiliser ces attributs ?

L'interface de base qui vous est proposée a deux avantages. Le premier est le fait qu'elle offre un aspect commun pour tous les jeux (cela évite des trop grandes différences dans la manière d'entrer dans un jeu, ou d'utiliser les touches...).

Le second avantage est également important : Cette interface permet aux déficients visuels d'adapter l'aspect du jeu à leur vision. C'est-à-dire qu'ils peuvent en changer les couleurs, les voix, la taille de la police, afin que le jeu soit le plus accessible possible pour eux.

Nos attributs sont mis à jour lorsqu'on charge un profil, ou lorsqu'on change les couleurs dans l'interface de configuration. En cela. Vous devez les utiliser. (Si vous ne les utilisez pas, c'est-à-dire, si vous décidez de coder « en dur » par exemple du noir sur du blanc, les couleurs de votre jeu ne pourront pas être évolutives).

4.2. Comment utiliser ces attributs ?

C'est très simple !

Vous devez tout d'abord respecter les 2 consignes suivantes :

* **Eviter d'insérer d'images**, privilégier des composants graphiques peuvent être utilisés (on ne pourra pas changer leurs couleurs...)

* **Eviter d'insérer des sons préenregistrés** (wave ou MP3)

La remarque sur le son est à respecter dans la limite du possible. Par exemple, si vous créez un jeu de puzzle musical par exemple, il faudra bien mettre des sons prédéfinis ! Mais pour les voix par exemple, évitez d'enregistrer vos voix. Utilisez de préférence SI VOX pour que les déficients visuels puissent modifier les voix et leur débit.

Vous devez utiliser nos attributs comme dans l'exemple de code suivant :



```
/* Titre du laby */
titre.setFont(new Font(Settings.police,1,Settings.taillePolice));

/* Couleur de fond du labyrinthe */
g.setColor(Settings.couleurFond);

/* Couleur du triangle */
g2.setColor(Settings.couleurObjet);

/* Voix de bienvenue */
voix.playText("Bienvenue");
```

Mais il ne faut surtout pas faire comme dans l'exemple suivant :



```
/* Titre du laby */
titre.setFont(new Font("Arial",1,36));

/* Couleur de fond du labyrinthe */
g.setColor(Color.BLACK);

/* Couleur du triangle */
g2.setColor(Color.RED);

/* Voix de bienvenue */
voix.playWav("../mesSons/bienvenue.wav");
```

Enfin, il est conseillé de déclarer vos couleurs et police une fois pour toute au début des classes dans lesquelles vous les utilisez afin de récupérer une seule fois les « settings ».

A noter que les instances des trois SI VOX sont créées dans la mainFrame.

Enfin le style de police qu'il est recommandé d'utiliser est contenu dans Settings. Il vous suffira en fait d'appeler « Settings.police ».

5. Où coder votre jeu ?

Avant de commencer cette section, il faut IMPERATIVEMENT avoir lu la section « IMPORTANT : Avant de commencer à coder ».

- L'interface graphique du jeu devra utiliser comme JPanel principal : **jeu.JeuPanel**
- Le noyau du jeu : où vous voulez (conseil : créer un nouveau package)

6. Comment ajouter des touches pour votre jeu ?

- **classesMeres**
 - *CommandeInterface* : déclarer les nouvelles actions
 - *ClasseAbstraite* : ajouter le corps par défaut des nouvelles actions
 - *MainFrame* : affecter une touche à chaque action
- **InterfaceJeu** : implémenter les actions des touches

7. Où coder les règles de votre jeu ?

Classes à modifier :

- **Regle**
 - *InterfaceRegle* : modifier le texte du rappel des touches spécifiques
 - *ReglePanel* : implémenter l'interface graphique des règles

Le *ReglePanel* doit décrire clairement et séparément

- le but du jeu
- les règles.

8. Qu'y a-t'il à votre disposition ?

Tous les attributs statiques contenus dans la classe *Settings*.
Les instances de *SIVOXDevint* initialisées dans *MainFrame*.